SINGLE AND MULTI-FRAME VIDEO QUALITY ENHANCEMENT

A Dissertation
Presented to
The Academic Faculty

By

Tarik Arici

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
August 2009

SINGLE AND MULTI-FRAME VIDEO QUALITY ENHANCEMENT

Approved by:

Russel M. Mersereau, Committee Chair
Professor, School of ECE
Georgia Institute of Technology

James Hamblen
Professor, School of ECE
Georgia Institute of Technology

Yucel Altunbasak, Advisor
Professor, School of ECE
Georgia Institute of Technology

Brani Vidakovic
Professor, Dept. of Biomedical Engineering
Georgia Institute of Technology

Ghassan AlRegib
Asst. Professor, School of ECE
Georgia Institute of Technology

Date Approved: April 27, 2009

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Devices for displaying information can be categorized as visual and tactile displays. Tactile displays are usually designed for the visually impaired and use mechanical parts to update an image. Visual displays are the most common displays. Common types of visual displays are analog and digital electronic displays, and projectors. Cathode ray tube (CRT) is an analog electronic display device invented by Karl Ferdinand Braun in 1897 [1]. CRTs are used as oscilloscopes, televisions, and video and computer displays. Using CRTs for television[1] was first described by a letter written to *Nature* by Alan Archibald Campbell-Swinton in 1911 [2]. This was the very first version of the television method, which was later refined in 1926 and 1928. CRTs were used in all television sets until the 1990s, when plasma display panels (PDPs) and liquid crystal displays (LCDs) became practical. PDPs and LCDs are lighter and thinner than CRTs, which makes them more convenient for consumers [3]. By the early 2000s, LCDs started to dominate the television and computer screen markets.

LCD technology was not able to compete with PDPs in sizes of 40” or larger because PDPs had lower costs. By 2004, 40” and 45” LCD televisions were widely available with competitive market prices. After 2004, major manufacturers announced the successful manufacture of large display sizes such as 82” and 102”. With their higher resolution and lower power consumption compared to PDPs, LCDs are rapidly growing in the display market. Although large size LCDs are more attractive to consumers, they have a number of disadvantages, as given below [3][4].

- LCDs usually have a lower contrast ratio compared to PDPs. LCDs leak some of the black light because of their nature of operation, which decreases the contrast, which is defined as the difference between black and white pixels.

[1] A telecommunication method for broadcasting and receiving images.

- With larger display sizes artifacts in the source video are more visible. Noise or compression artifacts such as ringing, blocking, and color bleeding are disturbing on large displays.

- LCDs produce sharp images only in their native resolution and have to scale images at non-native resolutions. Unfortunately, scaling introduces blurriness.

- Refresh rates in NTSC (60 Hz), PAL (50 Hz), and SECAM (50 Hz) formats cause a flicker problem, which requires increasing the refresh rate typically to 100 Hz. Also, increasing the refresh rate gives a crispy look to the video.

- Video source rate is almost always lower than the refresh rate to reduce transmission and storage costs. However, lower rates introduce problems such as motion judder and blurriness. Judder is observed as the unnatural motion of objects in the scene, and blurriness results because of the inability of the observer to track and focus on the moving object.

To improve the visual quality, input video must be enhanced before sending to the display. Therefore, our goal is to develop hardware-friendly low-complexity enhancement algorithms. Video quality enhancement methods can be classified in two main categories. Single frame methods are the first category. These methods have generally low computational complexity. Multi-frame methods combine information from more than one frame and require the motion information of objects in the scene to do so.

1.1 Single Frame Methods

We first concentrate on the contrast-enhancement problem by using both global (frame-wise) and local information derived from the image. Global contrast-enhancement techniques mostly make use of the histogram equalization method. We also use the image histogram and present a regularization-based histogram modification method to avoid problems that are often created by histogram equalization.

Next, we design a compression artifact reduction algorithm that reduces ringing artifacts, which is disturbing especially on large displays. Video sources are usually stored in a digital medium and suffer from compression artifacts. With the increasing use of high-definition television (HDTV), which uses compression in its storage and transmission systems, dealing with compression artifacts will be a necessity for a pleasing visual experience.

Furthermore, to remove the blurriness in the original video we present a non-iterative diffusion-based sharpening algorithm, which enhances edges in a ringing-aware fashion. The diffusion-based technique works on gradient approximations in a neighborhood individually. This gives more freedom compared to modulating the high-pass filter output that is used to sharpen the edges.

1.2 Motion Estimation for Multi-frame methods

Motion estimation enables applications such as motion-compensated noise reduction, frame-rate conversion, de-interlacing, compression, and super-resolution [5]. Compression is only concerned about the residual error and motion-vector coding to achieve the highest compression ratio possible. However, enhancement applications such as noise reduction, frame-rate conversion, and de-interlacing require *true* motion of image pixels.

Motion estimation is an ill-posed problem and therefore requires the use of prior knowledge on motion of objects. Objects have inertia and are usually larger then pixels or a block of pixels in size, which creates spatio-temporal correlation.

We design a method that uses temporal redundancy to improve motion-vector search by choosing bias vectors from the previous frame and adaptively penalizes deviations from the bias vectors. This increases the robustness of the motion-vector search when there are low-detailed regions, repeating patterns, rotations, and occlusions.

Compared to the temporal correlation in the motion-vector fields, the spatial correlation is more reliable because temporal correlation is difficult to use when the objects move fast

or accelerate in time, or have small sizes. Spatial smoothness is not valid across motion boundaries. We investigate using energy minimization for motion estimation and incorporate the spatial smoothness prior into the energy. By formulating the energy minimization iterations for each motion vector as the primal problem, we show that the dual problem is motion segmentation for that specific motion vector. This information from the dual problem is used to reliably enforce smoothness prior even under motion boundaries. We also study simplifications to reduce computational complexity, and heuristics for fast convergence.

CHAPTER 2

A HISTOGRAM MODIFICATION FRAMEWORK FOR IMAGE CONTRAST ENHANCEMENT

2.1 Introduction

Contrast enhancement has a crucial role in image processing applications such as military, medical image analysis, and object recognition. There are several reasons for an image/video to have poor contrast: the quality, settings, and limitations of the used imaging device, expertise of the operator, and the external conditions at the time of acquisition. These reasons result in under-utilization of the offered dynamic range. As a result, such images and videos typically do not reveal all the details of the captured scene, and may have a washed-out and unnatural look. Contrast enhancement targets to eliminate these problems and as a result is one of the most common image enhancement techniques used in image processing and computer vision. Contrast enhancement is commonly used in digital photography, medical imaging, remote sensing, and scientific visualization to obtain a more visually-pleasing or informative image or both. Typical viewers describe the enhanced images as if a curtain of fog has been removed from the picture [5].

Several contrast enhancement techniques have been introduced to improve the contrast of an image. These techniques can be broadly categorized into two groups: direct [6, 7] and indirect methods [8, 9]. Direct methods define a contrast measure and try to improve it. Indirect methods, on the other hand, improve the contrast through utilizing the under utilized regions of the dynamic range without defining a specific contrast term. Most of the methods in literature fall into the second group. Indirect methods can further be divided into several subgroups: (i) techniques that decompose an image into high and low frequency signals for manipulation, e.g., homomorphic filtering [10], (ii) histogram modification techniques [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21], and (iii) transform-based techniques [22, 23, 24, 25, 26]. Out of these three subgroups, the second group received

the most attention due to its straightforward and intuitive implementation qualities.

Contrast enhancement techniques in the second subgroup modify the image through some pixel mapping such that the histogram of the processed image is more uniform than that of the original image. Techniques in this subgroup either enhance the contrast globally or locally. If a single mapping derived from the image is used then it is a global method; if neighborhood of each pixel is used to obtain a local mapping function then it is a local method. Using a single global mapping derived from the image or its histogram cannot enhance the local contrast. However, if local information such as local histogram is used for histogram equalization, local contrast will also be enhanced [14, 17]. Local histogram is computed using a sub-image around the current pixel. A mapping is derived from the local histogram which is used to map the current pixel's intensity. The method presented in this paper is demonstrated as a global contrast enhancement (GCE) method, and can be extended to local contrast enhancement (LCE) using similar approaches.

One of the most popular GCE techniques is histogram equalization (HE). HE is an effective technique to transform a narrow histogram by spreading the intensity clusters in the histogram [27, 28], and it is adaptive since it is based on the histogram of a given image. However, HE without any regularization can cause extreme modifications on the histogram, i.e., considerably changing the input image and resulting in an unnatural look.

Methods for limiting the level of enhancement of an image have been proposed, most of which are obtained through modifications on HE. Bi-histogram equalization was proposed to reduce mean brightness change [11]. HE produces images with mean intensity that is approximately in the middle of the dynamic range. To avoid this, two separate histograms from the same image are created and equalized independently. The first is the histogram of intensities that are smaller than the mean intensity, the second is the histogram of intensities that are bigger than the mean intensity. A similar method called equal area dualistic sub-image histogram equalization (DSIHE) was proposed in which the two separate histograms were created using the median intensity instead of the mean intensity [12]. Although they

perform better than HE, these two techniques can not adjust the level of enhancement and are not robust to noise which becomes a problem when the histogram has high bins. Also, it should be noted that preserving the brightness does not imply preservation of naturalness. One method to deal with high histogram bins is low-pass filtering the histogram [13]. Another method proposes modifying the "cumulation function" of the histogram to adjust the level of enhancement [14], but is still sensitive to problems created by high histogram bins.

There are also unconventional approaches to the histogram-based contrast enhancement problem [15, 16]. Gray-level grouping (GLG) is such an algorithm that groups histogram bins and then redistributes these groups iteratively [15]. Although GLG can adjust the level of enhancement and is robust to high histogram bins, it is mainly designed for still images. Since gray-level grouping makes hard decisions on grouping histogram bins, and redistributing the bins depends on the grouping, mean brightness intensity in an image sequence can abruptly change in the same scene. This causes flickering which is one of the most annoying problems in video enhancement. Although a fast version of the algorithm is available, GLG's computational complexity is high for most applications.

Contrast enhancement techniques in the first and third subgroups use multi-scale analysis to decompose the image into different bands and enhance desired global and local frequencies [10, 22, 23, 24, 25, 26, 29, 30, 31]. These techniques are computationally complex but enable global and local contrast enhancement at the same time by enhancing the appropriate scales. In addition, they have limitations and are not fully automatic.

The aforementioned contrast enhancement techniques perform well on some images but they can create problems when a sequence of images is enhanced, or when the histogram has high bins, or when a natural looking enhanced image is strictly required. Computational complexity becomes an important issue when the goal is to design a contrast enhancement algorithm for consumer products such as Television. In addition, consumer electronics require the method to be controllable. The proposed framework gives the properties of the

enhancement mapping $T[x]$, for a natural looking enhancement and presents the enhancement problem as a regularized histogram approximation problem to handle issues arising from un-regularized histogram equalization. The proposed regularization terms are for adjusting the level of enhancement, noise robustness, and black/white stretching. The computational complexities of the regularization terms are discussed and a low-computational complexity algorithm handling the mentioned problems associated with HE is presented.

In the next section, contrast enhancement is explained. In Section III, the contrast enhancement using the proposed framework is explained in a progressive manner. Then, the low-complexity proposed method is presented in Section IV. Simulation results and discussion is presented in Section V. Finally, conclusion is provided in Section VI.

2.2 Contrast Enhancement

Histogram-based contrast enhancement techniques utilize the image histogram to obtain a single-indexed mapping $T[x]$ to modify intensities of image pixels. In HE and other histogram-based methods, mapping function is obtained from the histogram or its modified histogram, respectively [27]. HE finds a mapping to obtain an image with a histogram that is as close as possible to a uniform distribution to fully exploit the dynamic range. A histogram, $h[n]$, can be thought of as an un-normalized discrete probability mass function of the pixel intensities. The normalized histogram $p[k]$ of an image gives the approximate probability density function (pdf) of its pixel intensities. Then, the approximate cumulative distribution function (cdf), $c[k]$, is obtained from $p[k]$. The mapping function is a scaled version of this cdf. HE uses the image histogram to obtain the mapping function; whereas, other histogram-based methods obtain the mapping function of the modified histogram. The mapping function in the discrete form is

$$T[x] = \left\lfloor 2^B \sum_{k=0}^{x} p[k] + 0.5 \right\rfloor, \tag{2.1}$$

where B is the number of bits used to represent gray levels. Although the histogram of the processed image will be as uniform as possible, it may not be exactly uniform because of

the discrete nature of the pixel intensities.

It is also possible to enhance the contrast without using the histogram. Oldest tricks used in consumer-grade TV sets are black stretching and white stretching, which are very simple but effective techniques [5]. Black stretching makes dark pixels darker, while white stretching makes bright pixels brighter. This produces more natural looking black and white regions; hence, enhances the contrast of the image. Linear black and white stretching can be achieved by

$$T[x] = \begin{cases} x \times s_b & (x < b) \\ g[x] & (b \leq x \leq w), \\ x_{max} + (x - w) \times s_w & (w \leq x) \end{cases} \tag{2.2}$$

where b is the maximum intensity to be stretched to black and w is the minimum intensity to be stretched to white, $g[x]$ is any function mapping the intensities in between, and s_b, s_w are black and white stretching factors both of which are less than one.

2.3 Histogram Regularization

To fully exploit the available dynamic range, HE tries to create a uniformly distributed output histogram by using a cumulated histogram as its mapping function. However, HE often produces overly enhanced unnatural looking images. The problem with HE rises from large derivatives of T. To deal with this, the input histogram must be modified without giving up on its contrast enhancement potential. The modified histogram can then be accumulated to map input pixels to output pixels, similar to HE.

It is important to note that when the input distribution is already a uniform distribution, the mapping obtained from cumulating the input distribution is $T(x) = x$. Hence to lessen the level of enhancement that would be obtained by HE, the input histogram $\mathbf{h}$ can be modified to $\tilde{\mathbf{h}}$ in such a way that it is *closer* to a uniformly distributed histogram $\mathbf{u}$.

The modified histogram can be seen as a regularized approximation of the input histogram. The goal is to find a modified histogram $\tilde{\mathbf{h}}$ that is as close as desired to $\mathbf{u}$, and also

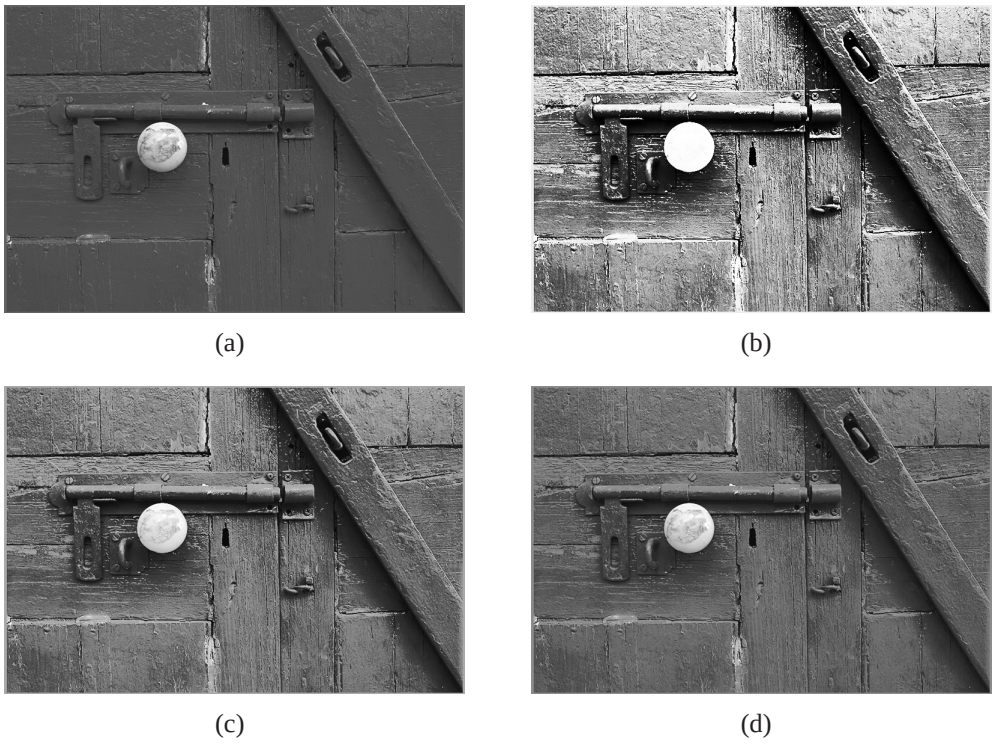


Figure 2.1. (a) Original image, (b) enhanced image with $\lambda = 0$, (c) enhanced image with $\lambda = 1$, (d) enhanced image with $\lambda = 2$

make the residual $\tilde{\mathbf{h}} - \mathbf{h}$ small. This modified histogram will be used to obtain the mapping function using (2.1). This is a convex bi-criterion problem, and can be formulated as a weighted sum of the two objectives as

$$\min ||\tilde{\mathbf{h}} - \mathbf{h}|| + \lambda||\tilde{\mathbf{h}} - \mathbf{u}||, \tag{2.3}$$

where $\mathbf{h}$, $\tilde{\mathbf{h}}$, and $\mathbf{u} \in R^{256\times1}$, and λ is a problem parameter. As λ varies over $(0, \infty)$ the solution of (2.3) traces out the optimal trade-off curve between the two objectives. HE obtained by the minimum value of λ correspond to the standard HE, and the maximum value of λ corresponds to not applying any equalization. Therefore, various levels of contrast enhancement can be achieved by varying λ.

2.3.1 Adjustable Histogram Equalization

An analytical solution to (2.3) can be obtained when the squared sum of the Euclidean norm is used, i.e.,

$$\min ||\tilde{\mathbf{h}} - \mathbf{h}||_2^{\ 2} + \lambda ||\tilde{\mathbf{h}} - \mathbf{u}||_2^{\ 2}, \tag{2.4}$$

which results in the quadratic optimization problem:

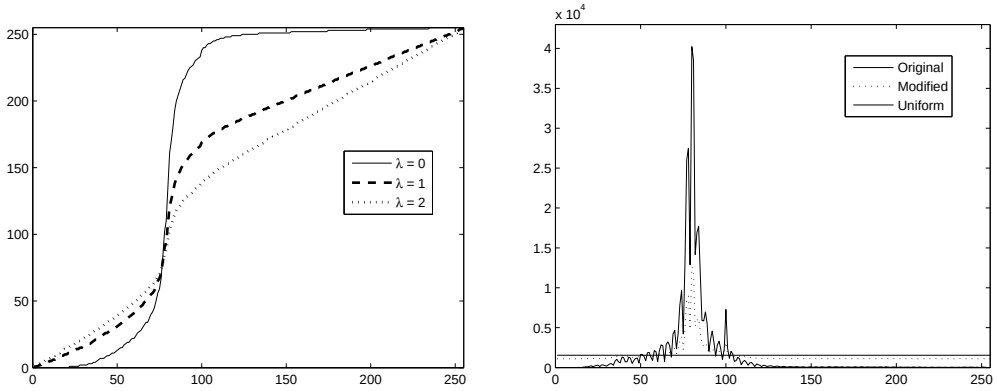
$$\min (\tilde{\mathbf{h}} - \mathbf{h})^T (\tilde{\mathbf{h}} - \mathbf{h}) + \lambda (\tilde{\mathbf{h}} - \mathbf{u})^T (\tilde{\mathbf{h}} - \mathbf{u}). \tag{2.5}$$

The analytical solution to this problem is

$$\tilde{\mathbf{h}} = \frac{\mathbf{h} + \lambda \mathbf{u}}{1 + \lambda}. \tag{2.6}$$

The modified histogram $\tilde{\mathbf{h}}$ turns out to be a weighted average of $\mathbf{h}$ and $\mathbf{u}$. By varying the weight λ, various levels of enhancement can be achieved.

An example image and enhanced image using modified histogram equalization with three different λ values (0, 1, 2) are shown in Fig. 2.1. When λ is zero, the modified histogram is equal to the input histogram, hence, the standard HE is applied. The resulting image is over-enhanced, with many unnatural details on the door and loss of all details on the doorknob. When λ is increased to one, the regularization term comes into account and the enhanced image looks more like the original image. For $\lambda = 2$, the level of enhancement is further decreased and the details on the doorknob are kept. In Fig. 2.2(a), the mappings for the three λ values are given. With increasing λ the mapping becomes more like a line with a slope of one, which is a mapping that does not change the input image. The fixed point observed around intensity value of 76 is a repelling fixed point[1]. Although the level of enhancement is decreased with increasing λ, the slope of the mapping at the fixed point, x^*, is still big. The slope of the mapping, $\dot{T}(x^*)$, as discussed in Theorem **??** determines the speed that the intensities in the enhanced image move away from the fixed point. This becomes an important issue especially in images with smooth background in which intensity differences in neighboring pixels look like noise.



(a) (b)

Figure 2.2. a) Mappings for 3 different λ values, b) Original histogram, modified histogram with $\lambda = 2$ and the uniform histogram

(a) (b) (c) (d)

Figure 2.3. (a) Original image, (b) enhanced image with $\gamma = 0, \lambda = 1$, (c) enhanced image with $\gamma = 0, \lambda = 3$, (d) enhanced image with $\gamma = 1000, \lambda = 1$

The problem of large $\dot{T}(x^*)$ arises from outliers in the input histogram. The original histogram given in Fig. 2.2(b) exhibits large bins and the modified histogram has also large bins for the corresponding intensities. This outlier sensitivity is observed because ℓ_2 norm assigns large penalty to large residuals and therefore is not robust to outliers. To remedy this problem, ℓ_1 norm can be used for the histogram approximation term in the objective while ℓ_2 norm is still used for the regularization term. Hence, the problem in (2.4) is changed to

$$\min ||\tilde{\mathbf{h}} - \mathbf{h}||_1 + \lambda ||\tilde{\mathbf{h}} - \mathbf{u}||_2{}^2, \tag{2.7}$$

To transform this mixed norm problem into a constrained quadratic programming problem, the first term can be expressed as a sum of auxiliary variables

$$\min \mathbf{t}^T \mathbf{1} + \lambda (\tilde{\mathbf{h}} - \mathbf{u})^T (\tilde{\mathbf{h}} - \mathbf{u})$$

[1]Please see ??? for a detailed discussion of repelling/attractive fixed points.

subject to

$$-\mathbf{t} \preceq (\tilde{\mathbf{h}} - \mathbf{h}) \preceq \mathbf{t},$$

where $\mathbf{t} \in R^{256\times 1}$ and represents the auxiliary variables[2], and $\mathbf{1} \in R^{256\times 1}$ is a vector of ones. However, this constrained quadratic programming problem has high computational complexity since there are 512 optimization variables.

Another way to deal with the outliers in the input histogram is to use one more regularization term to measure the smoothness of $\tilde{\mathbf{h}}$, which reduces the modified histogram's sensitivity to outliers.

2.3.2 Histogram Smoothing

To avoid outliers that leads to strong repelling fixed points, a smoothness constraint can be added to the objective. The gradient approximation of the histogram, i.e., $h[i] - h[i-1]$, can be used to measure its smoothness. A smooth modified histogram will tend to have less outliers since outliers are essentially abrupt changes in the histogram.

The gradient matrix $D \in R^{255\times 256}$ is bidiagonal

$$D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}$$

With the additional regularization term for smoothness, the optimal trade-off is obtained by minimizing

$$\min \|\tilde{\mathbf{h}} - \mathbf{h}\|_2^2 + \lambda\|\tilde{\mathbf{h}} - \mathbf{u}\|_2^2 + \gamma\|\mathbf{D}\tilde{\mathbf{h}}\|_2^2, \tag{2.8}$$

[2] $\preceq$ symbol denotes vector/componentwise inequality.

The solution of this three-criterion problem is

$$\tilde{\mathbf{h}} = ((1+\lambda)\mathbf{I} + \gamma\mathbf{D}^T\mathbf{D})^{-1}(\mathbf{h} + \lambda\mathbf{u}). \tag{2.9}$$

While (2.6) finds a weighted average of **h** and **u**, (2.9) further smoothes this weighted average to avoid outliers. It is possible to show that the first term in (2.9), that is, $S^{-1} = ((1+\lambda)\mathbf{I} + \gamma\mathbf{D}^T\mathbf{D})^{-1}$ is in fact a low-pass filtering operation on the averaged histogram by explicitly writing $S = ((1+\lambda)\mathbf{I} + \gamma\mathbf{D}^T\mathbf{D})$ as

$$S = \begin{bmatrix} 2\gamma+(1+\lambda) & -2\gamma & 0 & 0 & \cdots \\ -2\gamma & 4\gamma+(1+\lambda) & -2\gamma & 0 & \cdots \\ 0 & -2\gamma & 4\gamma+(1+\lambda) & -2\gamma & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{2.10}$$

As can be clearly seen, S is a tridiagonal matrix.

Then, from [32], S^{-1} is given by

$$\begin{aligned} S^{-1}(i,j) &= k*(2\gamma)^{|i-j|}+(c_1*v_1^{256-|i-j|}+c_2*v_2^{256-|i-j|} \\ &\quad +c_3*(v_1*v_2)^{256-|i-j|}), \end{aligned} \tag{2.11}$$

where k, c_1, c_2, and c_3 are constants, and v_1 and v_2 are the eigenvalues of the 2×2 matrix

$$\begin{bmatrix} 4\gamma+(1+\lambda) & -4\gamma^2 \\ 1 & 0 \end{bmatrix},$$

which always have positive eigenvalues.

It can be seen from (2.11) that each row of S^{-1} is a zero-phase low-pass filter. Hence, a regularization term for smoothness corresponds to low-pass filtering the original histogram. This explains heuristic histogram low-pass filtering approaches investigated in literature as in [13].

To illustrate the performance of histogram smoothing, the image given in Fig. 2.3(a), which is captured from a compressed video stream is enhanced using adjustable histogram equalization with and without histogram smoothing. Both Fig. 2.3(b) and Fig. 2.3(c) adjusts

the level of enhancement with $\lambda = 1$ and $\lambda = 3$, respectively. After enhancement, both of them are corrupted with artifacts that are observed as black grain noise around the text. These artifacts arise from the strong repelling fixed point in the mapping created by the outlier bins of the original histogram. The ringing artifact pixels that have intensities less than the background pixels are mapped to much darker intensities. Histogram smoothing with $\gamma = 1000$ solves this problem as can be seen in Fig. 2.3(d). The mappings for the corresponding enhanced images are given in Fig. 2.4. $\dot{T}(x)$ at the outlier bin intensity has been successfully decreased with histogram smoothing.

Although histogram smoothing is successful in avoiding outlier histogram bins, it has a shortcoming. For a real-time implementation S^{-1} has to be computed for each image as γ needs to be adjusted depending on the magnitude of the outlier histogram bins. Even though there are fast algorithms for inverting tridiagonal matrices requiring only $O(7n)$ arithmetic operations [**?**] as opposed to $O(n^3/3)$, it is still unacceptable because of the application at hand. This renders the algorithm not hardware implementable. Instead of using (2.9), a low-pass filtering on the histogram can also be performed. But the number of taps and the transfer function must also be adaptive similar to the form given in (2.11) for S^{-1}. Another approach that is less computationally complex is to use a weighted error norm for the approximation error $\tilde{\mathbf{h}} - \mathbf{h}$, which is to be described next.

2.3.3 Weighted Histogram Approximation

Histogram outliers occur because of the existence of large number of pixels with exactly the same intensity values compared to neighboring intensity values. This condition makes the derivative of the mapping at that intensity also large. This results in mapping of a very narrow range of pixel values to a wider range of pixel values. Hence, it causes contouring and grainy noise type artifacts in uniform regions. Large number of pixels with exactly the same intensity are due to big smooth areas in the image. Hence, measuring the average local variance of all pixels with the same intensity can be used to weight the approximation error, $\tilde{\mathbf{h}} - \mathbf{h}$. Histogram approximation error at the corresponding bin will be weighted with

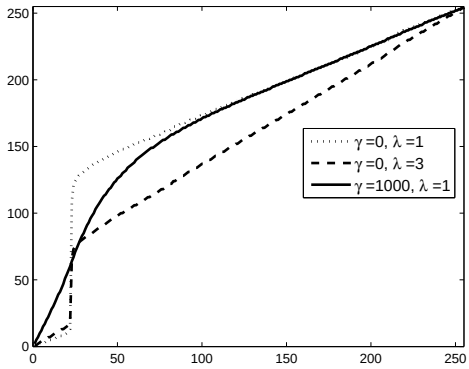


Figure 2.4. Mappings for the enhanced images given in Figure 2.3

a small weight. Therefore, the modified histogram bin will not follow the input histogram's outlier bin to minimize the approximation error. The objective function with the weighted approximation error is

$$\min(\tilde{\mathbf{h}} - \mathbf{h})^T \mathbf{W}(\tilde{\mathbf{h}} - \mathbf{h}) + \lambda(\tilde{\mathbf{h}} - \mathbf{u})^T(\tilde{\mathbf{h}} - \mathbf{u}). \tag{2.12}$$

where $\mathbf{W} \in R^{256\times256}$ is a diagonal error weight matrix, and $W(i,i)$ measures average local variance of pixels with intensity i. The solution of (2.12) is

$$\tilde{\mathbf{h}} = (\mathbf{W} + \lambda\mathbf{I})^{-1}(\mathbf{W}\mathbf{h} + \lambda\mathbf{u}) \tag{2.13}$$

and it is computationally simpler than (2.9). Since the first term is a diagonal matrix, taking matrix inverse is avoided, i.e., simple division operation is used to find its inverse.

Fig. 2.5 shows the weighted histogram approximation and histogram smoothing for comparison. The grain noise type artifact pixels around the text is avoided in both methods. The mappings for the two methods is given in Fig. 2.6. The derivative of the mapping corresponding to smooth background pixels has further been reduced. However, the mapping is not as smooth as histogram smoothing since no smoothing is performed on the modified histogram.

2.3.4 Black and White Stretching

Black and white (B&W) stretching is one of the oldest image enhancement techniques used in Television sets. B&W stretching maps predetermined dark and bright intensities

(a) (b)

Figure 2.5. (a) Histogram smoothing with $\gamma = 1000, \lambda = 1$, (b) Weighted Approximation with $\lambda = 1000$

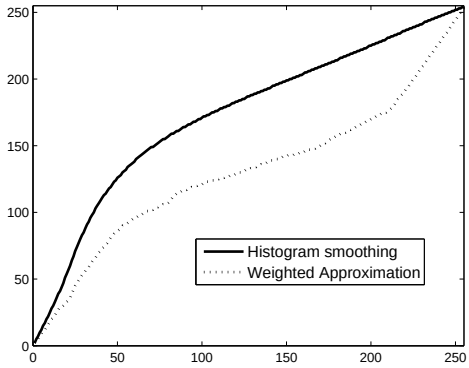


Figure 2.6. Mappings for the enhanced images given in Figure 2.5

to darker and brighter intensities, respectively. To incorporate B&W stretching into histogram modification, where the intensity range for B&W stretching is $[0, b]$ and $[w, 255]$ respectively, the modified histogram $\tilde{\mathbf{h}}$ must have small bins for the corresponding intensity ranges. Since the length of the histogram bins determines the contrast between the mapped intensities, by decreasing the histogram bin length for $[0, b]$ and $[w, 255]$, the mapping obtained by accumulating the modified histogram will have a smaller derivative for these two intensity ranges.

An additional regularization term for B&W stretching can be added to one of the objective functions presented in previous subsections (e.g., adjustable histogram equalization equation given in (2.5))

$$\min(\tilde{\mathbf{h}} - \mathbf{h})^T(\tilde{\mathbf{h}} - \mathbf{h}) + \lambda(\tilde{\mathbf{h}} - \mathbf{u})^T(\tilde{\mathbf{h}} - \mathbf{u}) + \alpha\tilde{\mathbf{h}}^T\mathbf{I}^B\tilde{\mathbf{h}}, \tag{2.14}$$

where $\mathbf{I}^B$ is a diagonal matrix. $\mathbf{I}^B(i,i) = 1$ for $i \in \{[0, b] \cup [w, 255]\}$, and the remaining

(a) (b) (c) (d)

Figure 2.7. (a) Original image, (b) enhanced image with $\gamma = 1000, \lambda = 1$, (c) enhanced image with $\gamma = 100, \lambda = 1, \alpha = 5$, (d) mappings for the two images

diagonal elements are zero. The solution to this minimization problem is

$$\tilde{\mathbf{h}} = \left((1+\lambda)\mathbf{I} + \alpha\mathbf{I}^B\right)^{-1}(\mathbf{h} + \lambda\mathbf{u}). \tag{2.15}$$

In Fig. 2.7, histogram smoothing with and without B&W stretching is given. With the more natural look of the black and white in the image, the contrast has greatly improved. Black stretch intensity range is $[0, 20]$ and white stretch intensity range is $[200, 255]$ with α set to 5. The mapping as given in Fig. 2.7(d) clearly shows B&W stretching and the smooth transition to non-stretching region.

2.4 A Low-complexity Histogram Modification Algorithm

In this section a low-complexity histogram modification algorithm is presented. The low-complexity algorithm deals with histogram outliers, performs B&W stretching, and adjusts the level of enhancement adaptively so that the dynamic range is maximally utilized without amplifying the noise visibility and degrading the natural look of the image. The proposed algorithm does not perform any division operation.

Using histogram smoothing or weighted histogram approximation is computationally complex when considering the scarce memory and area resources in the hardware. Histogram smoothing requires either solving (2.9) or explicit low-pass filtering with adaptive filter length and transfer function. On the other hand, weighted approximation with solution given in (2.13) requires division operation.

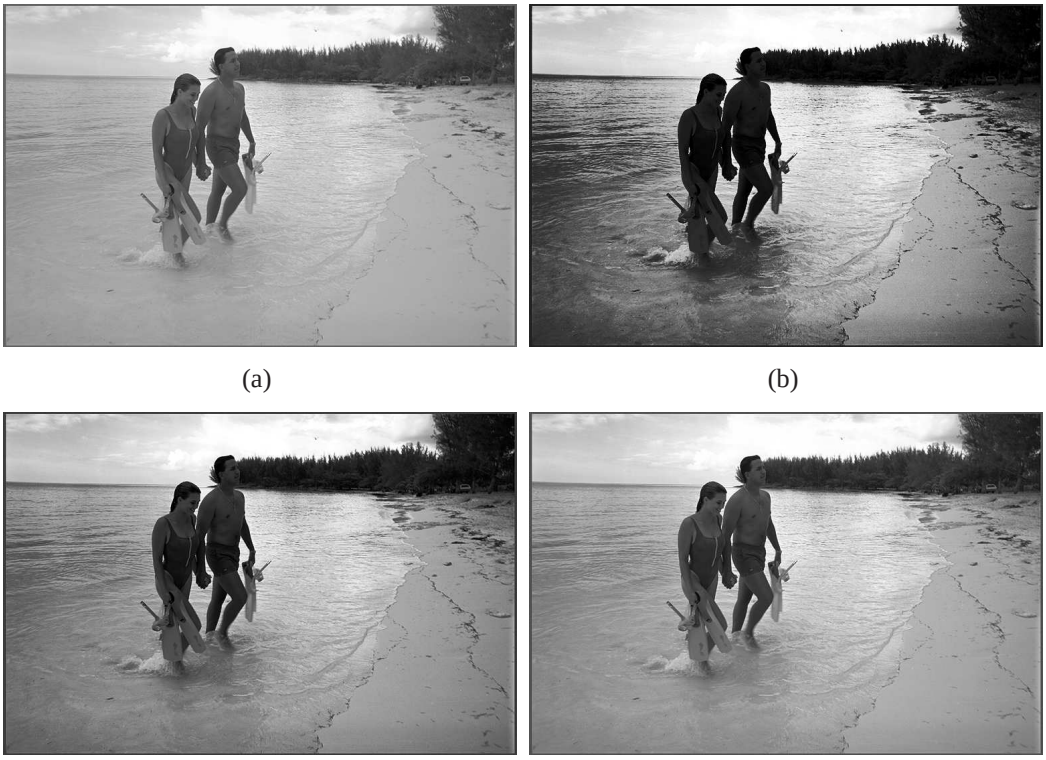
2.4.1 Histogram Creation

To deal with histogram outliers in a simple way, instead of smoothing or weighting the input histogram, the way of obtaining the input histogram can be modified. Since histogram outliers are created because of large number of pixels that have the same intensity, these pixels almost always come from smooth areas in the image. Even if they do not, it will not be a problem because grainy noise type artifacts are visible in smooth areas. Hence, histogram creation can be modified by counting pixels that have some level of contrast with their neighbors. This way, a histogram can be regarded as a discrete conditional probability $h[x] = p(x|contrast)$, i.e., $h[x]$ is the estimated probability of a pixel having intensity i given that it has a contrast with its neighbors. Performing histogram equalization on $p(x|contrast)$ instead of $p(x)$, will enhance the contrast but not the noise since the dynamic range is best utilized for pixels that have some level of contrast with their neighbors.

To obtain the histogram, the local variation of each pixel can be used to decide if a pixel has sufficient contrast with its neighbors. One efficient way of achieving this for hardware simplicity is to use a horizontal variation measure by taking advantage of the row-wise pixel processing architecture, which is available in common video processing hardware. A horizontal one-lagged difference operation is a high-pass filter, which will also measure noise. On the other hand, a horizontal two-lagged difference operation is a band-pass filter which will not pass high-frequency noise signals. Histogram is created using pixels with a two-lagged difference that has a magnitude larger than a given threshold (*steps 5, 6, 7*). The number of pixels included in the histogram is also counted.

2.4.2 Adjusting The Level Of Enhancement

As described in Section 2.3.1, it is possible to adjust the level of histogram equalization to achieve natural looking enhanced images. The modified histogram is a weighted average of the input histogram **h** and the uniform histogram **u**, as given in (2.6). The contribution of the input histogram in the modified histogram is $\kappa = \frac{1}{1+\lambda}$. The level of histogram equalization should be adjusted depending on the input image's contrast. Low contrast images



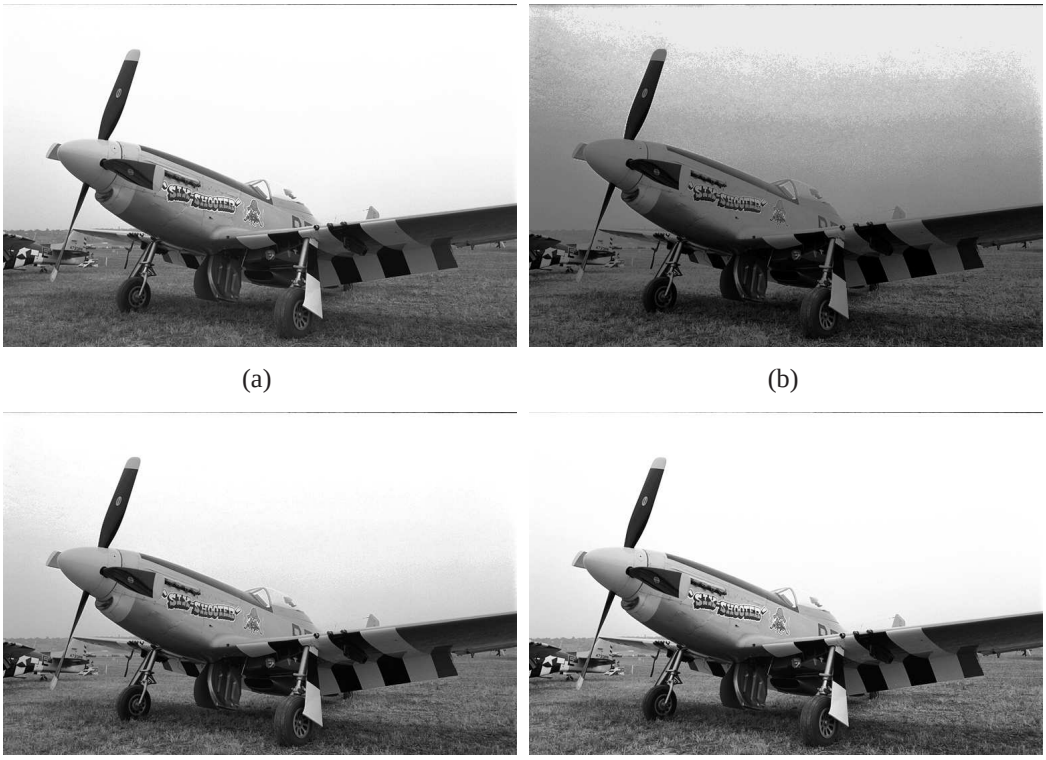
(a) (b)

(c) (d)

Figure 2.8. (a) Original image, (b) HE image, (c) WTHE image, (d) proposed algorithm

have narrow histograms and with histogram equalization contouring and noise can be created. Therefore, κ is computed to measure the input contrast using the aggregated outputs of horizontal two-lagged difference operation (*step 4*). Afterwards, κ is non-linearly modulated and normalized to the range $[0, 1]$ (*step 11*). To ensure that **h** and **u** have the same normalization, **u** is obtained using the number of pixels that are included in the histogram (*step 12*). $u_{\min}$ is used to ensure that very low bin regions of the histogram will not result in very low slope in the mapping function; it will increase the slope in these regions, resulting in increased-utilization of dynamic range.

B&W stretching is performed using (2.15) (*step 17*). Parameters b, w, and α can be adapted with the image content. b and w is usually derived from the histogram as the minimum and maximum intensities. For noise robustness, b should be chosen as the minimum intensity that is bigger than some predefined number of pixels' intensities, w can be chosen



(a) (b)

(c) (d)

Figure 2.9. (a) Original image, (b) HE image, (c) WTHE image, (d) proposed algorithm

similarly. It is a good practice to impose limits on b and w. The stretching parameter should also be adapted with image content. For dark images white stretching can be favored, while for bright images black stretching can be favored. α may also depend on the input image's contrast.

2.5 Results and Discussion

Proposed algorithm has been successfully tested on a variety of test images and video sequences. A few of the results are shown in this paper. Fig. 2.8 and Fig. 2.9 show the original test images and their corresponding contrast enhanced versions. Their mapping functions are shown in Fig. 2.10(a) and Fig. 2.10(a), respectively. The proposed algorithm is compared with a recent state-of-the-art global contrast enhancement algorithm in literature, known as weighted thresholded HE (WTHE), presented in [18], in which they compare

Algorithm 1: GCE Histogram Modification Algorithm

Input: Input image: f,
B&W stretch parameters: b, w, and α
Output: Modified histogram: $\tilde{\mathbf{h}}$

Initialize κ;
foreach *row m* **do**
  foreach *column n* **do**
    $\kappa = \kappa + |f[m,n] - f[m,n-2]|$;
    if $|f[m,n] - f[m,n-2]| > Threshold$ **then**
      $++h[f[m,n]]$;
      $++count$;
    end
  end
end
Normalize κ;
$u = \min\{count/256, u_{\min}\}$;
foreach *bin i* **do**
  if $b < i < w$ **then**
    $\tilde{h}[i] = (1-\kappa)u + \kappa h[i]$;
  else
    $\tilde{h}[i] = \frac{1}{1+\alpha}[(1-\kappa)u + \kappa h[i]]$;
  end
end

it against the algorithms presented in [11, 19, 20] and show their algorithm's superiority. Both WTHE and our proposed method show similar visual quality on many of the images we tested. However, that is not always the case. Hence, images included in this paper are selected among the ones that cause different visual quality.

Histogram equalized images result in the best utilization of the dynamic range of the pixel values for maximum contrast. However, this often does not mean the resulting image is better in terms of visual quality. This situation is also observed with images in Fig. 2.8(b) and Fig. 2.9(b). Undesired artifacts become more prominent, and amplified nature of noise degrades the quality of the image resulting in an unnatural look. WTHE and the proposed algorithm on the other hand offers a controllability of the contrast enhancement. Since the histogram of the proposed algorithm is formed from the conditional probability, it does not have very high bins resulting from uniform regions; hence, the proposed method does not

produce artifacts as HE and WTHE, which are caused by having high bins that cause high slope in the mapping function. Even though WTHE thresholds high bin values to prevent its undesired effect, it does not produce as pleasing results as the proposed algorithm does. One other situation HE and WTHE introduces artifacts when the dynamic range of the original image is shrunk from either one or both ends. In either case, the resulting image is either darkened and/or brightened more than necessary. The proposed algorithm, on the other hand, avoids this situation through the use of mixing of conditional histogram and $u_{\min}$ as explained in line 14-18 of the Algorithm 1. By modifying the histogram, the proposed method improves the natural-look of the image substantially compared to HE and WTHE.

Fig. 2.8(b) is the histogram equalized image of 2.8(a). The contrast of the image is maximized at the expense of the amplified noise, and image artifacts. The resulting artifacts are mostly in the darker regions, which is also evident from Fig. 2.10(a). Darker regions become darker, and very bright region gets even brighter. WTHE reduces the effect of HE. However, the resulting image still has some flavor of HE: bodies of two people, and the trees are still darker and the resulting image Fig. 2.8(c) is not as visually pleasing as Fig. 2.8(d). As can be clearly seen, the mapping function in the region around 175 has a very steep curve resulting in a stretching of a very narrow region into a wider region; range of [150-180] is getting mapped to [60-180]. These pixel values are mostly due to the sand and some part of the sea. The pixel values of the two bodies are around 60 and the pixel values of the trees are around 40. The mapping functions for both HE and WTHE are mapping these values into darker pixel values. This is caused by the histogram of the original image having very few pixel values below 40. However, in the proposed algorithm this situation is prevented by filling very low bins adaptively with $u_{\min}$ as illustrated in Algorithm 1. Hence, the contrast enhanced image obtained by the proposed method is visually more pleasing than HE and WTHE.

Fig. 2.9(b) is the histogram equalized image of 2.9(a). HE image, again, looks very

unnatural. Especially, the dominance of the sky region results in a very big slope in the mapping function around the pixel value of 250, which results in mapping of range [250-256] into [150-256]. Unnatural look of the HE image is lessened using WTHE. However, it is not alleviated completely. Graininess in the sky still exist in the regions close to the plane. The proposed algorithm, on the other hand, produces a good visual quality result; there is no graininess in the sky and the contrast of the grass is improved compared to HE and WTHE result. The success of the proposed method in this type of images is, again, due to the use of the conditional histogram. Big uniform regions in an image cause corresponding bins in the histogram to be very high compared to other bins. However, conditional histogram avoids having very high bins. This feature is controlled adaptively by the variable κ in the algorithm. If an image contains large smooth regions, then the effect of histogram is lessened so that the resulting image preserves the smoothness and does not introduce visual artifacts. On the other hand, if there is no dominant smooth region in an image, then the effect of κ is increased to increase the contrast.

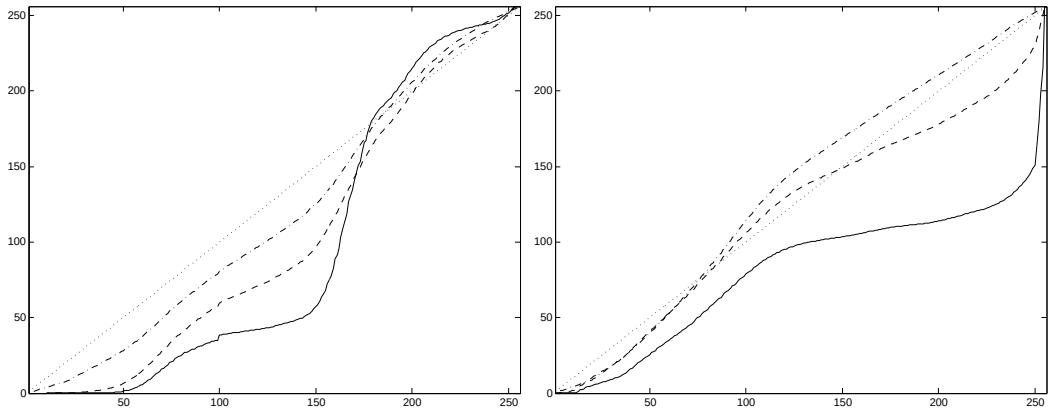


Figure 2.10. (a) mappings of Fig 2.8, (b) mappings of Fig 2.9. Solid line indicates the HE mapping, dashed line indicates the WTHE mapping, dash-dotted line indicates the proposed method, and the dotted line indicates the ideal no change mapping.

2.6 Conclusions

A general framework for image contrast enhancement is presented, and a low-complexity algorithm suitable for video display applications is proposed. The presented framework enables the adjustment of contrast enhancement through carefully designed regularization terms. Hence, the contrast of the image/video can be improved without introducing visual artifacts that decrease the visual quality of an image and cause it to have an unnatural look.

To obtain a real-time implementable and low-complexity algorithm, the proposed method avoids cumbersome calculations and bandwidth consuming operations. The experimental results show the effectiveness of the algorithm in comparison to other contrast enhancement algorithms. Obtained images are visually pleasing, artifact free, and natural looking. A very important feature of the proposed algorithm is that it does not introduce flickering, which is very important for video applications.

The proposed method is very powerful in terms of being controllable, and adaptive to different images. It offers a level of controllability, and adaptivity through which all different levels of contrast enhancement between full contrast enhancement, i.e., HE, and no contrast enhancement can be achieved.

CHAPTER 3

LOCAL CONTRAST ENHANCEMENT USING RECURSIVE FILTERS

3.1 Introduction

Global contrast-enhancement (GCE) methods target the problems such as improper lighting conditions that manifest themselves in a global fashion. On the other hand, local contrast-enhancement (LCE) methods target the visibility of local details in the image. Since both methods produce images with higher contrast, the enhanced images generally look more appealing than their corresponding originals.

Two well-known local-contrast enhancement methods are adaptive histogram equalization (AHE) [33] and adaptive contrast enhancement (ACE) [34][35]. AHE algorithms find local mappings using local histograms. Although AHE improves contrast, its computational complexity may not be acceptable for real-time applications. A bi-linear interpolation technique is presented in [33] for a block-based AHE. Another disadvantage of the AHE methods is that they often over-enhance the image by creating so-called *contrast objects* that were not visible in the original image. The enhanced image often does not look natural [14].

ACE methods consist of two main steps: (i) low-pass filtering of the image and (ii) enhancing the details by utilizing the low-pass filtered image. Preserving the edges is critical since the human visual system is sensitive to edges [36]. Increasing the contrast around edges without degrading the edge quality improves visibility and produces better looking images. Low-pass filtering of the images is done using either finite impulse response (FIR) or infinite impulse response (IIR) filters. FIR filters have been found to be more acceptable than IIR filters because of their easier implementation and handling. Also, stability is not an issue for FIR filters. IIR[1] filters are more difficult than FIR filters to understand and handle,

[1]Note that throughout this work, we use the terms IIR and recursive exchangeably.

and their stability is not guaranteed. Since the fundamental law of algebra applies only to one-variable polynomials and cannot be applied to multi-variable polynomials, analyzing stability is much harder, if not impossible for 2-D IIR filters, [37][38]. A low-pass IIR filter has to be zero-phase, but it is impossible to design a zero-phase filter with a single recursive filter. However, zero-phase can be achieved by adding/multiplying a recursive filter with its complex conjugate. Our filters are first-order recursive filters, which are discussed in Section 3.2.

3.2 Adaptive Contrast Enhancement

In conventional ACE algorithms the enhanced image $y(m,n)$ is obtained from the input image $f(m,n)$ as

$$y(m,n) = g(m,n) + [1 + \gamma(m,n)][f(m,n) - g(m,n)], \tag{3.1}$$

where $g(m,n)$ is the local mean, $\gamma(m,n)$ is the enhancement gain, m is the row number, and n is the column number.

Our filter finds the local mean by averaging two opposite direction recursive filters. For the 1-D case our recursive filters operate on a line, whereas for 2-D filters they operate on the entire image. The gain function is designed to suppress noise visibility in smooth regions. We use a gain function designed in our previous work [39].

3.2.1 1-D Recursive Filter

The local mean $g(m,n)$ at row m and column n is the output of the recursive filter, which is the average of two different filter outputs given by

$$g(m,n) = \frac{g^{+}(m,n) + g^{-}(m,n)}{2}, \tag{3.2}$$

where $g^{+}(m,n)$ and $g^{-}(m,n)$ are the outputs of the two opposite direction filters that run horizontally on a single row. The first filter runs from left to right and is referred to as the forward running filter. The second filter runs from right to left and is, similarly, referred

to as the backward running filter. From this point forward, we will refer to them as the forward and backward filters, respectively. The two filters are single pole IIR filters at any given pixel location. The input-output relationship for the forward filtered $g^+(m,n)$ is

$$g^+(m,n) = \lambda(m,n)g^+(m,n-1) + [1-\lambda(m,n)]f(m,n), \tag{3.3}$$

where $\lambda(m,n)$ is the edge adaptive delay coefficient. The relationship for the backward filtered $g^-(m,n)$ is similarly,

$$g^-(m,n) = \lambda(m,n)g^-(m,n+1) + [1-\lambda(m,n)]f(m,n). \tag{3.4}$$

. The adaptation of $\lambda(m,n)$ to the edge information is crucial for preventing the smoothing of edges in the un-sharp masked image. Considering that $\lambda(m,n)$ is the weight of the previous output, a stronger $\lambda(m,n)$ increases the low-pass characteristic of the filter. Hence, when an edge is encountered, $\lambda(m,n)$ must be decreased so that the edge will be preserved in the output. The edge signal (E) we use is $|g_F(m,n-1)-f(m,n)|$ for the forward filter, and $|g_B(m,n+1)-f(m,n)|$ for the backward filter. Both of the edge signals are the differences between the original pixel value and the previous filter output. Using these edge signals, $\lambda(m,n)$ is obtained using

$$\lambda(m,n) = e^{-\alpha E}, \tag{3.5}$$

for the forward filter, and similarly for the backward filter. As can be observed from (3.5), strong edges reduce $\lambda(m,n)$ more, hence the low-pass characteristic of the filter is lessened. Typical α values are in the range of 0.01 to 0.04.

From (3.3) and (3.4) the frequency response of the forward and backward filters at a locality with $\lambda(m,n) = \lambda$ are given as below

$$H^+(z) = \frac{1-\lambda}{1-\lambda z^{-1}} \tag{3.6}$$

$$H^-(z) = \frac{1-\lambda}{1-\lambda z}. \tag{3.7}$$

Here, we implicitly assume that λs for the two filters are equal since ideally edge information at the same locality must be the same. Stability of the two transfer functions is guaranteed for $|\lambda| < 1$ which is satisfied using (3.5).

The z-transform of the local mean filter on the unit circle using (3.2), (3.6), and (3.7) is obtained as below

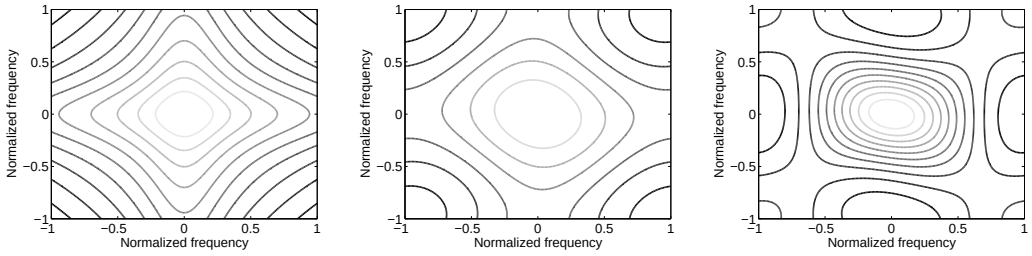
$$H(z) = (1 - \lambda)\frac{1 - \lambda Re\{z\}}{1 - 2\lambda Re\{z\} + \lambda^2}, \tag{3.8}$$

which has a zero phase. Thus, the 1-D local mean filter applied row-wise and given by (3.2) does not shift the input signal.

It is worthwhile to note that the above 1-D filter for the local mean is an exponential smoothing filter with impulse response

$$h[n] = ke^{-(\alpha E)|n|}, \tag{3.9}$$

where $k = (1 - e^{-\alpha E})$. Using the edge signal E, smoothing power of the filters is adapted spatially. This way with only a single tap for the delayed output, different levels of smoothing is achieved easily. To attain the same level of smoothing, a non-recursive Gaussian filter would require roughly $N = 4\sigma$ taps [37].



(a) Separable filter (b) Non-separable 4-neighbors filter (c) Non-separable 8-neighbors filter

Figure 3.1. Transfer functions of the 2-D recursive filters

3.2.2 2-D Separable Recursive Filter

2-D separable recursive filter is composed from 1-D filters running in the horizontal and vertical directions back to back.

$$\begin{aligned} H(z_1, z_2) &= H_x(z_1)H_y(z_2) \\ &= \frac{(H_x^+(z_1) + H_x^-(z_1))}{2} \frac{\left(H_y^+(z_2) + H_y^-(z_2)\right)}{2} \end{aligned} \tag{3.10}$$

Image is filtered in the horizontal direction first and then the resulting image is filtered in the vertical direction. Since $H_x(z_1)$ and $H_y(z_2)$ are the averages of forward and backward 1-D filters and $H_x(z_1)$ is cascaded to $H_y(z_2)$ bounded-input-bounded-output stability is again satisfied with $\lambda < 1$. The overall transfer function for the 2-D separable filter is as below

$$H_s(z_1, z_2) = \frac{1 - \frac{\lambda_a}{2}(z_1^{-1} + z_1)}{(1 - \lambda_a z_1^{-1})(1 - \lambda_a z_1)} \frac{1 - \frac{\lambda_b}{2}(z_2^{-1} + z_2)}{(1 - \lambda_b z_2^{-1})(1 - \lambda_b z_2)} \tag{3.11}$$

Since diagonal frequency components are filtered two times compared to horizontal and vertical components, the filters transfer function is not isotropic as can be seen in Fig. 3.1(a). This feature of the 2-D separable filter's transfer function favors more of the diagonal frequency components in the detail image. Therefore, amplified detail image added back to the un-sharp masked image to produce the enhanced image may cause false diagonal edges which is most disturbing on the smooth areas.

Next, we look at the 2-D filter, which recurses on the 4-neighbors to improve the isotropy of the transfer function.

3.2.3 2-D Filter Using 4-Neighbors

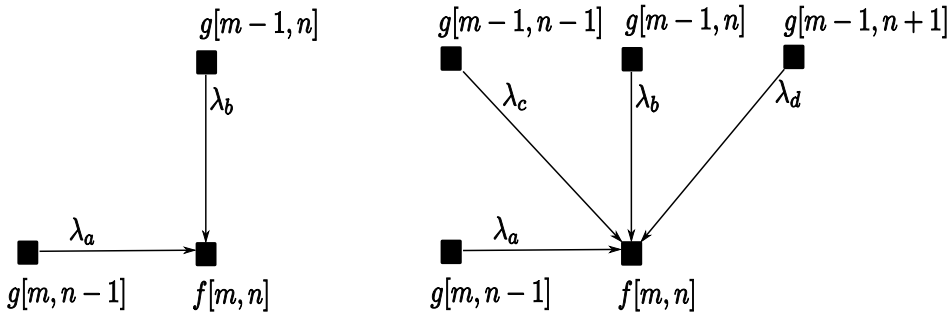
In this case, each of the forward and backward filters use 2 of the 4-neighbors for the recursion. 4-neighbors are the closest neighbors to the current pixel in the horizontal and vertical directions. Fig. 3.2(a) shows the 2 neighbor pixels used in the forward recursion, which are previously outputted. To guarantee zero-phase filtering, the overall filter response is again averaged from the forward and backward filter outputs. Transfer function for the

forward filter is

$$H_4^+(z_1, z_2) = \frac{1 - \lambda_a - \lambda_b}{1 - \lambda_a z_1^{-1} - \lambda_b z_2^{-1}}, \tag{3.12}$$

where $|\lambda_a| + |\lambda_b| < 1$ for stability [38]. To enable a low-complexity implementation with a small sized look-up table we tighten the constraint by using (3.5) and normalizing it with $\frac{1}{2}$ for both λ_a and λ_b. Hence, one LUT can be used for both and they can independently take values without needing to check for stability.

Backward filter is the complex conjugate of the forward filter on the unit circle. It uses the remaining 2 pixels of the 4-neighborhood.



(a) 4-Neighbors case (b) 8-Neighbors case

Figure 3.2. Neighbors used in the forward filters

The overall transfer function is

$$H_4(z_1, z_2) = \frac{1}{2}\left[H_4^+(z_1, z_2) + H_4^-(z_1, z_2)\right], \tag{3.13}$$

which is given in Fig. 3.1(b). As expected, transfer function's isotropy have improved compared to the separable filter.

3.2.4 2-D Filter Using 8-Neighbors

To improve the transfer function further, we include the remaining neighbors in the 8-neighborhood into the recursion. Hence, all 8-neighbors contribute to the output. Fig. 3.2(b) shows the neighbors for the forward filter. Similarly, this filter is also composed of forward and backward filters so that the zero-phase condition is satisfied. The filter for the forward

case is given as

$$H_8^+(z_1, z_2) = \frac{1 - \lambda_a - \lambda_b - \lambda_c - \lambda_d}{1 - \lambda_a z_1^{-1} - \lambda_b z_2^{-1} - \lambda_c z_1^{-1} z_2^{-1} - \lambda_d z_1 z_2^{-1}} \tag{3.14}$$

where the stability condition is for any $|z| = 1$

$$1 + a^2 > b^2 + c^2 + d^2 + cd(z^2 + z^{-2}) - 2(bd + bc)Re\{z\} \tag{3.15}$$

derived using a simplified procedure to check the conditions of Huang's stability theorem [40]. To avoid having to check this stability condition on each adaptation of delay coefficients (i.e. λs), we tighten the constraint by using (3.5) for each λ and normalize it with $\frac{1}{4}$. Hence, one LUT can be used for all four of them and they can independently take values without needing to check for stability.

Backward filter is the complex conjugate of the forward filter on the unit circle, and it uses the remaining 4 neighbors not shown in the figure. Again, the overall filter response is the average of both the forward and backward filters.

$$H_8(z_1, z_2) = \frac{1}{2}\left[H_8^+(z_1, z_2) + H_8^-(z_1, z_2)\right]. \tag{3.16}$$

As can be seen from Fig. 3.1(c), the isotropy of the transfer function is further improved compared to the earlier cases.

3.2.5 Computational Complexity Analysis

Our 2-D filters becomes more isotropic as we move from 2-D separable to 2-D 8-neighbors filter, which improves the visual quality of the enhanced image. Improvement in the visual quality brings along the increased computational complexity.

The delay coefficient (λ) of the IIR filters does not have to be computed each time since it is determined by (3.5) and the edge signal that is input to this function is always rounded to an integer between 0 and 255. Hence, for λ and the enhancement gain one look-up table (LUT) for each of them can be used. The memory needed for the LUTs is 256 bytes for λ LUT, around 20 bytes for enhancement gain LUT, one frame-store for the forward filter's output and an additional single register for backward filter's output.

Number of computations for a pixel in each filtering is discussed below so that the complexity-visual quality trade-off analysis can be made. 1-D filter requires 2 additions and 1 LUT for Eq. 3.1, and 2 multiplication, 4 additions, 1 bit shift and 1 LUT for forward and backward filters, totaling 2 multiplications, 6 additions, 1 bit shift and 2 LUTs.

2-D separable filter requires twice of 1-D filter for calculation of μ since it is cascade of 1-D filters in the horizontal and vertical. So, in total 2-D separable filter requires 4 multiplications, 10 additions, 2 bit shift and 2 LUTs.

Non-separable 4-neighbors filter requires 4 multiplications, 8 additions, 1 bit shift and 2 LUTs for forward and backward filters. Hence, in total it requires 4 multiplications, 10 additions, 1 bit shift and 2 LUTs. Similarly, non-separable 8-neighbors filter requires 8 multiplications, 18 additions and 2 LUTs.

3.3 Experiment Results

We have tested the images and videos obtained from the proposed filtering methods on an 94cm LCD TV for visual quality. The pertinent specifications of the TV are as follows: a resolution of 1280×768 along with a contrast ratio of 800:1. Some of the observations presented below are difficult to make from the given figures.

From Fig. 3.3(b), we can see that 1-D filter enhances the horizontal contrast most noticeable on the vertical iron bars of the fence. 2-D separable filter does vertical contrast enhancement in addition to the horizontal, which can be seen on the horizontal bars of the fence and shadows of the bus' glass window. Also, the vertical enhancement is evident on the pink flowers. The amount of vertical and horizontal contrast enhancement is more with the 2-D filter that uses 4-neighbors. This is easier to see on the rock and the vertical wood column in Fig. 3.4(c) compared to Fig.3.3(b). The contrast improvement with the 2-D filter using 8-neighbors is less compared to the improvement using 4-neighbors. However, using 8-neighbors performs better in terms of the enhancement quality in smooth regions. This is because of its transfer function's better isotropy. Since all the frequency components

are treated equally, no component is favored relative to others. However, with decreasing isotropy the transfer function filters out the diagonal frequency components more, and thus the detail image includes more of the diagonal components. The human eye can see this discrimination of the frequencies especially in smooth areas as false edges and objects in diagonal directions. This is most evident in the 2-D non-separable filter as expected.

(a) (b) (c)

(d) (e)

Figure 3.3. (a) First frame of Bus sequence enhanced with different filters: (b) Enhanced with 1-D filter, (c) Enhanced with 2-D separable filter, (d) Enhanced with 2-D filter using 4-neighbors, (e) Enhanced with 2-D filter using 8-neighbors

3.4 Skin-Aware Local Contrast Enhancement

Human intelligence is highly trained to recognize human faces and skin. Starting from the early infancy phases, humans learn to recognize other people with their faces [41], and communicate indescribable feelings and thoughts with simple facial mimics. The human face is therefore of great interestas Cicero said, everything is in the face. Special care must be taken for face and skin regions of an image before displaying. Faces in local contrast enhanced images may sometimes look unnatural although other regions look visually more

(a) (b) (c)

(d) (e)

Figure 3.4. (a) First frame of Tempete sequence enhanced with different filters: (b) Enhanced with 1-D filter, (c) Enhanced with 2-D separable filter, (d) Enhanced with 2-D filter using 4-neighbors, (e) Enhanced with 2-D filter using 8-neighbors

attractive than the original. There is a clear need for detecting skin and non-skin regions and applying a lighter level of local contrast enhancement to skin regions, if not some smoothing.

Pixel-based skin detection algorithms utilize the fact that skin colors are clustered in the color space [42, 43, 44]. Although there are small variations with respect to race and illumination conditions, skin color lies inside a definite shaped color space region (i.e. skin locus) [45]. Variations on the skin color due to race and lightning conditions are explained by intensity variations in the chrominance components. Therefore, color space based skin color detection methods are robust to lightning conditions. Another reason to discard the luminance component is to decrease the computational complexity through dimensionality reduction.

Skin color modeling based skin detection methods have 90% true positive, and 20% false positive rate on average depending on the model and (to some extent) the color space

used [46]. Since our goal is to discriminate skin pixels for a lighter level of enhancement, the true positives compared to the false positives are given more importance. To increase the performance of skin detection algorithms, more information is needed. One such information is the skin color homogeneity based on the assumption that skin regions in general consist of smooth skin patches. This information is especially useful in nudity detection [47]. With the help of the homogeneity assumption true positive and false positive rates can be improved moderately (e.g. to 96%, and 4.5% respectively [44]).

A low-complexity human skin aware local contrast enhancement (sLCE) method is presented. sLCE utilizes the likelihood of a pixel to be a skin pixel in order to modulate the amplification gain of the detail image for local contrast enhancement. The skin color distribution is modeled with a Gaussian distribution on the YCbCr color space. Contrary to pixel-based skin detection algorithms that utilize thresholds, a soft discrimination between skin and non-skin pixels by using the skin likelihood is used. This way, unwanted false edges caused by false classification is prevented. Furthermore, to deal with isolated non-skin pixels in skin regions (or the opposite), edge information is used to impose correlation between skin likelihoods of neighboring pixels. Thus, a skin likelihood map consists of contiguous skin regions corresponding to actual skin regions. Experiment results show that SLCE produces natural looking face and non-facial skin regions in the enhanced image while keeping the same level of enhancement on non-skin regions.

Skin color models can be divided into two groups in general: parametric and non-parametric models. Non-parametric models are mainly histogram-based models learned from training data sets. They require memory and may not generalize depending on the representativeness of the training set. However, parametric modeling interpolates the training set and generalizes better. The interpolation is a desired feature for the objective since a soft discrimination between skin and non-skin pixels is wanted.

3.4.1 Gaussian skin color model

Modeling skin color distribution with a Gaussian model is done before in [48, 49, 50]. The joint probability density function of the color vector, **c**, of a pixel given that it is a skin pixel is defined as:

$$p(\mathbf{c}|s=1) = \frac{1}{2\pi|\mathbf{\Sigma}|^{1/2}} e^{\frac{1}{2}(\mathbf{c}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{c}-\mu)}, \tag{3.17}$$

where $s = 1$ indicates a skin pixel, μ is mean vector, and $\mathbf{\Sigma}$ is covariance matrix. The Gaussian parameters estimated in [43] for the YCbCr color space are used

$$\mu = \begin{bmatrix} \mu_b \\ \mu_r \end{bmatrix} = \begin{bmatrix} 108.15 \\ 152.00 \end{bmatrix} \tag{3.18}$$

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_{\mathbf{b}}^2 & \sigma_{\mathbf{br}} \\ \sigma_{\mathbf{br}} & \sigma_{\mathbf{r}}^2 \end{bmatrix} = \begin{bmatrix} 55.77 & -58.66 \\ -58.66 & 85.27 \end{bmatrix} \tag{3.19}$$

Since linear combinations of Gaussian variables are also Gaussian variables, to find the Gaussian parameters for other colorspaces,

such as the YUV for analog composite video, all one needs to do is to transform the above estimates using the transformation matrix T between the two colorspaces (i.e. the transformed estimates are $T\mu, T\Sigma T^T$).

Using Bayesian formula the posterior probability of a color vector to come from a skin pixel (i.e. the skin likelihood) is

$$p(s=1|\mathbf{c}) = \frac{p(\mathbf{c}|s=1)p(s=1)}{p(\mathbf{c})} \tag{3.20}$$

The prior probability of skin $p(s = 1)$ will be a constant depending on the type of the image or video. To save computation, color vectors are assumed to be uniformly distributed on the color space (i.e. all colors are equally likely with $p(\mathbf{c}) = \frac{1}{224^2}$). Hence,

$$p(s=1|\mathbf{c}) = k\, p(\mathbf{c}|s=1). \tag{3.21}$$

$k = 1$ is used in the following discussions and the experimental results.

An example image and its skin likelihood map is given in Fig. 3.5(a) and Fig. 3.5(b). Isolated non-skin pixels in skin regions and skin pixels in non-skin regions can be seen. This is partly due to the Gaussian model's failure (e.g. the non-skin pixels on the face) and use of the color information only to detect the skin. For example pixels on the edge's of the shirt have skin color because slow chrominance transitions imposed by the composite analog video broadcast standards (e.g. NTSC, PAL) cause hue change artifacts and in this case the hue changes from shirt's red color to skin color on the edges. To deal with the Gaussian model's failure, the edge information is used to impose spatial correlation between the skin likelihoods.

As described in Section 3.2.1, λ of the recursive LCE filter (a number between 0 and 1) is updated with the edge information using (3.5) such that lambda decreases with the edge strength. To deal with isolated skin likelihoods (SLs), the skin likelihood is correlated in a locality with a recursive estimation that adapts the amount of correlation according to the edge strength. No correlation across the edges is imposed.

Since λ is between 0 and 1, the probability of edge is assumed to be $p(E = 1) = (1 - \lambda)$; then, the probability of no edge is $p(E = 0) = \lambda$. Then, SL in a locality, $\bar{p}(s = 1)$, can be written in terms of its conditional probabilities:

$$\bar{p}^{new}(s = 1) = \bar{p}^{old}(s = 1|E = 0)p(E = 0) + p(s = 1|\mathbf{c})p(E = 1), \tag{3.22}$$

where $\bar{p}^{new}$ is the updated local SL using the old local SL ($\bar{p}^{old}$) and current pixel's SL. In other words, the old local SL is used if there is no edge and current pixel's SL is used if there is an edge. To decrease the estimation variance of the local SL, a weighted average using the current pixel's SL is calculated as

$$\bar{p}^{old} = \frac{7}{8}\bar{p}^{old} + \frac{1}{8}p(s = 1|\mathbf{c}). \tag{3.23}$$

Substituting (3.23) and edge probabilities in (3.22) gives

$$\bar{p}^{new} = \frac{7}{8}\lambda\bar{p}^{old} + (1 - \frac{7}{8}\lambda)p(s = 1|\mathbf{c}). \tag{3.24}$$

An example map for the correlated skin likelihood is given in Fig. 3.5(c).

Using this skin likelihood, the enhancement gain can be modulated so that a skin pixel will be enhanced less than a non-skin pixel. This way, the natural look of the face can be preserved while other parts of the image are enhanced. This modulation can be done using the updated local SL ($\bar{p}^{new}$) (3.24) in (3.1) as below

$$\begin{aligned} y(m,n) &= \mu(m,n) + [1 + (1 - \bar{p}^{new})g(m,n)] \\ &\quad [x(m,n) - \mu(m,n)]. \end{aligned} \tag{3.25}$$

Thus, amplification of the local detail will decrease with its skin likelihood. Other modulation functions can also be used such as the square root of skin likelihood if less amount of skin enhancement is desired. Furthermore, for a better visual quality one can smoothen skin regions by contracting the deviation from local luminance mean instead of amplifying it with a gain. This can simply be done by making the gain negative for highly likely skin regions as below

$$\begin{aligned} y(m,n) &= \mu(m,n) + [1 + \frac{k - \bar{p}^{new}}{k} g(m,n)] \\ &\quad [x(m,n) - \mu(m,n)]. \end{aligned} \tag{3.26}$$

where the used gain modulation function ($\frac{k-\bar{p}^{new}}{k}$) is negative for $\bar{p}^{new} > k$ and 1 for $\bar{p}^{new} = 0$.

3.4.2 Experiment Results for Human Skin

Video captured from an NTSC broadcast is used for performance evaluation. Fig. 3.5(c) shows that by correlating the SLs, isolated likelihoods are lessened and smoothed. Fig. 3.5(c) gives an imaged enhanced using LCE only. Unnatural looking skin regions can be seen on the nose, sides of the mouth and on the right side of the neck. These artifacts on the face and the neck are almost removed using SALSA as shown in Fig. 3.5(e) and Fig. 3.5(h). It is important to note that there is no loss of enhancement on non-skin regions. Smoothing the skin regions looks more natural as given in Fig. 3.5(f) and Fig. 3.5(i), which are smoothed using $k = 0.75$. However, this comes with a cost and the level of enhancement on the other image part is not as strong as SALSA without smoothing.

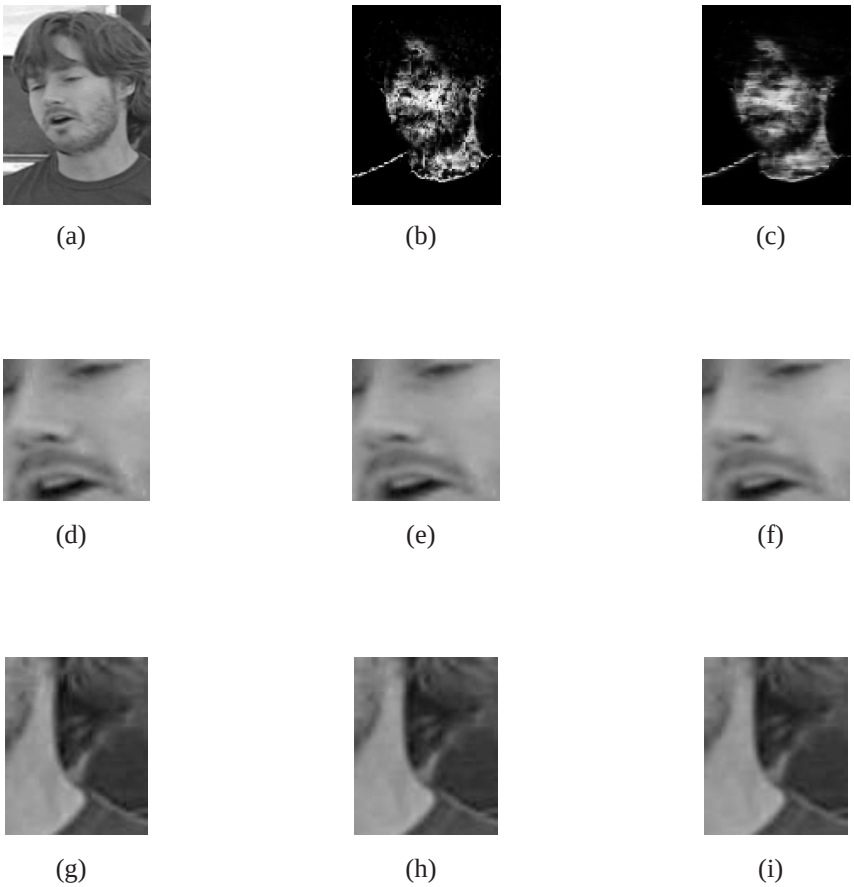


Figure 3.5. (a) Original image, (b) Uncorrelated skin likelihood map, (c) Correlated skin map, (d) LCE of face, (e) SALSA of face, (f) SALSA with smoothing of face (k=0.75), (g) LCE of neck, (h) SALSA of neck, (i) SALSA with smoothing of neck (k=0.75)

3.5 Conclusions

Compared to FIR filters, recursive filters can achieve desired transfer functions with much fewer coefficients enabling low-complexity algorithms which can be used in consumer electronics products such as Television. While using recursive filters special care must be taken to ensure that the transfer function has zero phase and the output is stable in the bounded-input-bounded-output sense. A family of exponential smoothing filters using recursive filters to obtain edge-preserving low-pass images as an unsharp-mask of the original image for contrast enhancement is presented. However, a uniform enhancement of all pixels in the image can sometimes produce unnatural images especially of human faces. To deal with this problem, modulating the enhancement gain with the skin likelihood is proposed.

Spatial correlation is imposed on the likelihoods to reduce isolated skin and non-skin pixels. Furthermore, a light level of smoothing for the face is also proposed while enhancing other parts of the image.

CHAPTER 4

RINGING ARTIFACT REDUCTION

4.1 Introduction

Our de-ringing method is illustrated in Fig. 4.1. Ringing artifact-free image is obtained from the original image via smoothing with the edge-preserving recursive filter. In the ringing artifact-free image, texture is also smoothed out along with the ringing artifacts. Hence, smoothed pixels must be used when the original pixels are ringing artifact pixels. The ringing likelihood of each pixel, which is computed from the edge map, is used to modulate the *alpha* blending of the two image pixels.

Recursive filters operate on previously computed outputs. A two-dimensional one-lagged recursion can be implemented using horizontal, vertical, and diagonal neighbors. Using all one-lagged neighbors, the transfer function becomes more isotropic as discussed in Section 3.2. The recursion when using the horizontal and vertical neighbors is

$$g[m,n] = \lambda_a g[m,n-1] + \lambda_b g[m-1,n] + (1-\lambda_a-\lambda_b) f[m,n], \tag{4.1}$$

where $f[m,n]$ and $g[m,n]$ are the input image pixel and the filter output at location $[m,n]$, and λ_a and λ_b are recursion coefficients. Also,

$$|\lambda_a| + |\lambda_b| < 1, \tag{4.2}$$

for stability. To achieve a zero-phase transfer function, complex conjugate filter working

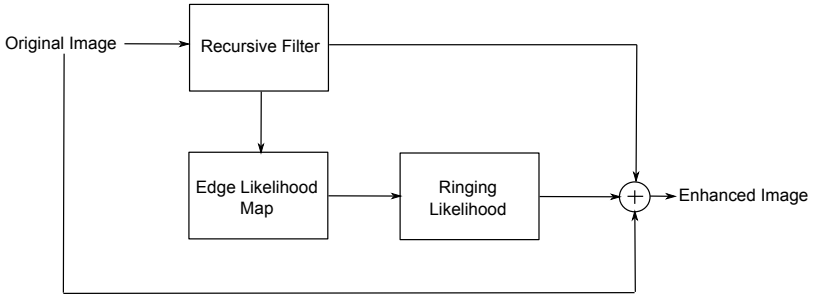


Figure 4.1. Block diagram for the proposed ringing artifact reduction method

in the opposite direction is also employed. Moreover λ_a and λ_b is again adapted with edge

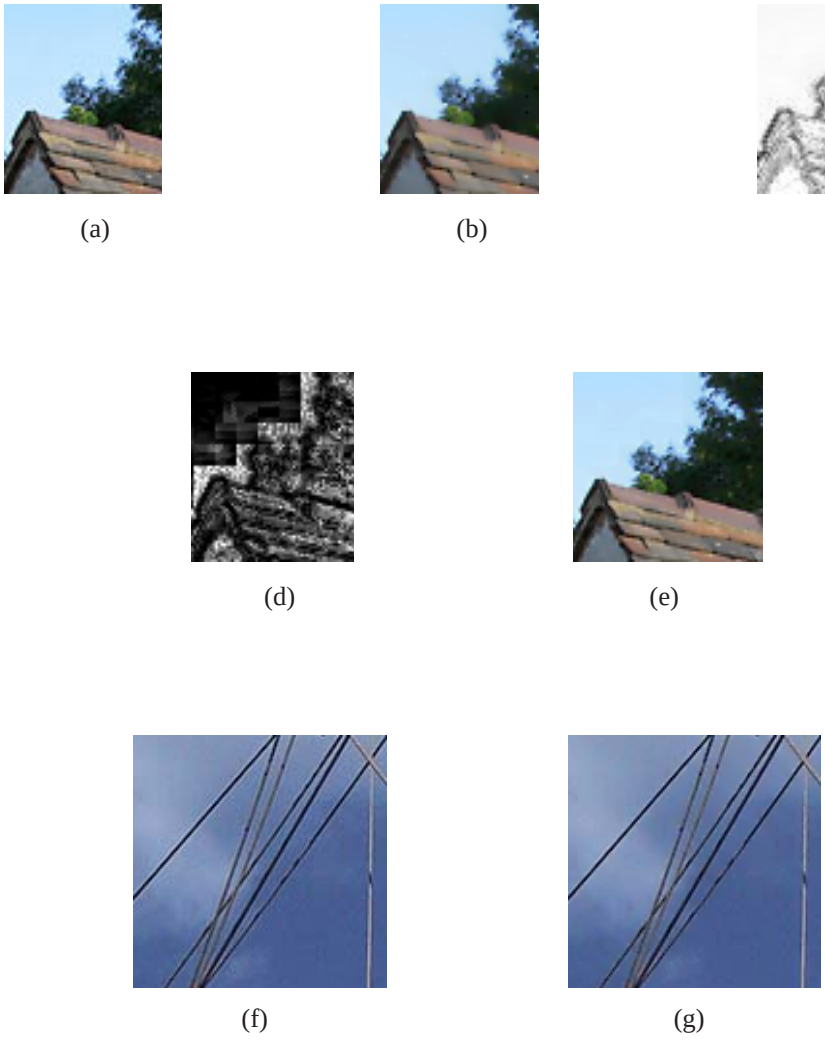


Figure 4.2. (a) Original image with ringing artifacts, (b) image smoothed with the 2-D recursive filter, (c) edge likelihood map, (d) ringing likelihood map, (e) result of the proposed method, (f) an example image with ringing artifacts, (g) result of the proposed method.

In Fig. 4.2a-b, a JPEG compressed image and its smoothed version using the 2-D recursive filter is given. In the smoothed image, edges are preserved but the ringing artifacts and the texture is filtered out. To keep the details but remove the ringing artifact pixels, original image and smoothed image can be *alpha* blended by carefully changing α for each pixel.

As discussed in Section 4.1, ringing is created because of the DCT quantization. Therefore, coding blocks with more non-zero DCT coefficients are more prone to ringing. Since

DCT blocks with edges have a larger number of non-zero DCT coefficients, they are prone to ringing. Ringing is most visible in smooth areas around strong edges. It is easier for the human eye to detect the DCT noise which is contrasted with the smooth regions. Interestingly, ringing is masked by details in the image. It may even look like a true detail that enhances the visual quality. In Fig.4.2a, ringing artifacts in the roof region is less disturbing than the ones in the sky region. Therefore one needs to deal with ringing in smooth areas around edges. As a result, ringing likelihood of a pixel is closely related with the edges in the image. Fortunately, the 2-D recursive smoothing filter inherently produces edge information because it uses this information to adjust its smoothing amount. Since λ_a and λ_b are inversely related with the vertical/diagonal and horizontal/diagonal edges respectively,

and their sum is less than one (see Eq.4.2), edge likelihood $p^E[m,n]$ of a pixel can be defined as

$$p^E[m,n] = 1 - \lambda_a[m,n] - \lambda_b[m,n]. \tag{4.3}$$

In Fig.4.2c edge likelihood map of the image is shown, in which $p^E[m,n] = 1$ corresponds to black, and $p^E[m,n] = 0$ corresponds to white. It can be seen that ringing pixels have small edge likelihoods compared to the true edge pixel's edge likelihood.

4.2 Ringing Likelihood Estimation

As discussed above, the likelihood of a visible ringing in a pixel is related with its edge likelihood, and the presence of a strong edge in its coding block. Ringing likelihood ratio *LR* of a pixel is defined as

$$LR = \frac{p^{*E}[m^*,n^*]}{p^E[m,n]}, \tag{4.4}$$

where $p^{*E}[m^*,n^*]$ is the edge likelihood of the strongest edge located at $[m^*,n^*]$ in coding block **B** which is obtained by

$$p^{*E}[m^*,n^*] = \max_{[m,n]\in\mathbf{B}} p[m,n]. \tag{4.5}$$

LR is small if the current pixel is an edge, and large if it is not an edge and there exists a strong edge in its coding block.

Another way to view Eq. 4.4 is to think of LR as a

likelihood ratio in a hypothesis testing problem. The null hypothesis H_0 that the pixel at $[m^*, n^*]$ is the strongest edge pixel in $\mathbf{B}$, is tested against the alternative H_1 that the current pixel at $[m, n]$ is the strongest. Hence, LR is always greater than one. H_1 imposes the restriction that $[m^*, n^*] = [m, n]$. It often proves to be that case that under H_0

$$\chi^2(m) \approx 2 \ln LR, \tag{4.6}$$

which states that $2 \ln LR$ is Chisquare distributed with m degrees of freedom (m is one in this case)[51]. Probability of LR to be less than $LR_{observed}$ shows how likely is H_0 against H_1. Hence, ringing likelihood $p_R[m, n]$ of a pixel can be defined as

$$p^R[m, n] = p(LR < LR_{observed}). \tag{4.7}$$

Ringing likelihood map of the image in Fig.4.2a is given in Fig.4.2d, in which $p^R[m, n] = 1$ corresponds to white and $p^R[m, n] = 0$ corresponds to black (opposite of the edge likelihood map). It can be seen that ringing pixels in the sky have large $p^R[m, n]$, while pixels on the roof have low $p^R[m, n]$.

4.3 Optimal Alpha Blending

An optimal *alpha* blending of the original image and the artifact-free smoothed image can be obtained using $p^R[m, n]$ of each pixel.

Each pixel $f^\dagger[m, n]$ in the decompressed source image is a random variable given by

$$f^\dagger[m, n] = I[m, n]g[m, n] + (1 - I[m, n])f[m, n], \tag{4.8}$$

where $I[m, n]$ is one if $[m, n]$ is a ringing artifact pixel, and zero otherwise. The *alpha* blended image $f^\ddagger[m, n]$ is computed by

$$f^\ddagger[m, n] = \alpha g[m, n] + (1 - \alpha)f[m, n]. \tag{4.9}$$

The error $\epsilon[m, n]$ between $f^\dagger[m, n]$ and $f^\ddagger[m, n]$ is

$$\epsilon[m, n] = (I[m, n] - \alpha)g[m, n] + (\alpha - I[m, n])f[m, n]. \tag{4.10}$$

For an unbiased estimator of $f^{\dagger}[m,n]$, $E(\epsilon[m,n])$ is set to zero, yielding

$$\alpha = E(I[m,n]) = p^{R}[m,n]. \tag{4.11}$$

4.4 Experiment Results

In image quality enhancement techniques, MSE-based performance measures do not really reflect the performance. Therefore, the results are visually examined and are very encouraging in terms of subjective visual quality inspection. Fig. 4.2f shows a JPEG compressed image with heavy ringing artifacts. Since the edges (ropes) are surrounded by smooth areas (sky), ringing artifacts are highly visible. The result of the proposed algorithm is given in Fig. 4.2g. Almost all ringing artifacts are successfully removed from the picture without degrading the quality of the edges.

CHAPTER 5

DIFFUSION BASED NON-ITERATIVE SHARPENING

5.1 Introduction

The visual quality of an image strongly depends on the quality of its edges. Edge information plays a fundamental role in the human visual system for extracting structural information in a scene [52]. In the literature, there are quality assessment techniques that are based on edge quality [53]. Enhancement methods for improving the edge properties become a significant component for consumer products. Among many image enhancement techniques, image sharpening algorithms try to make rapid intensity transitions in the edges. By doing so, sharpening, in effect, adds high-frequency components to the image, which makes it more pleasing to the eye.

There are numerous sharpening algorithms available in the literature, but, because of its low computational complexity and lower memory requirements, un-sharp masking (UM) has become the preferable one for display enhancement in consumer products. The UM methods have two basic and common processes. The first one is called un-sharp masking of the image, which is attained by low-pass filtering the original image. Second, the detail mask that has high-frequency components of the image will be obtained by subtracting the un-sharp mask from the original image. The enhanced image is produced by amplifying the detail mask and then adding it back to the un-sharp mask.

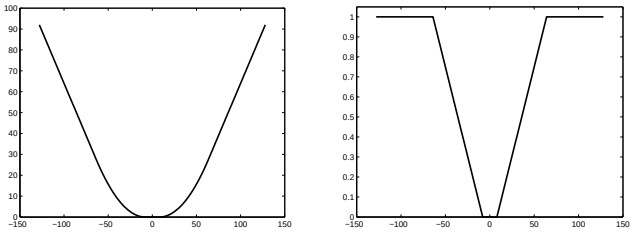
Most of the time, UM techniques are implemented directly by the equivalent high-pass filter that outputs the high-frequency component [54]. However, amplifying high-frequency components creates over/under shooting artifacts and boosts noise around the edges. To reduce noise amplification and over/under shooting artifacts in UM techniques, various methods are proposed [55][56]. All of these methods try to modulate the output of the high-pass filter so that compared to conventional UM, noise and over/under shooting performance is improved. Nevertheless, modulating the output of high-pass filter modulates each pixel

within the mask in the same way, which in return, may reduce the sharpening performance of these classes of algorithms. To clarify this claim, consider a noisy region in an image. We can expect that the UM algorithms will try to reduce the magnitude of the high-pass filter within and around these noisy regions. But, a weak intensity edge (*i.e.*, blurry edge) in this noisy region will not be sharpened as a result of this modulation. Therefore, while these algorithms aim to avoid the over/under shooting effect together with the noise amplification, a trade-off that reduces the sharpening performance comes into play. In this work, we propose an approach to relax this trade-off by modulating the contrast of each pixel with respect to the center pixel in the mask using a backward diffusion-based sharpening algorithm.

Backward diffusion[1] is not a stable process, and it creates oscillations around the edges [57]. One solution to avoid oscillations is to terminate the process after a limited number of iterations. Another approach is to actively smooth out small gradients using a forward and backward diffusion simultaneously to avoid introducing oscillations at the firsthand [58]. This inevitably smoothes weak intensity details in the image too. Therefore, instead of applying a forward diffusion, we choose to stop the diffusion after a limited number of iterations. To avoid boosting noise in smooth areas and creating over/under shooting artifacts, we present the problem as an objective-maximization problem. The objective function is a modified version of Huber's min-max norm of the local gradients. We further weight the objective function with the ringing likelihood of each pixel so that ringing artifact pixels contribute less to the objective and hence are not sharpened. By using the equivalence of the objective-maximization problem with the diffusion [59], we can formulate sharpening as an anisotropic diffusion process which is then approximated with a non-iterative method.

Next, we will present the image sharpening as an objective maximization problem. Our objective is to increase the contrast between pixels; most preferably the ones residing across the edges. The contrast between pixels is derived from the norm of their gradients.

[1] It is called as 'backward' since it works against the gradient direction.



(a) (b)

Figure 5.1. (a) $\rho(x)$ (b) $\rho^{'}(x)$ with $T_1 = 8$ and $T_2 = 64$.

We will present this modified norm and its necessary properties to achieve a satisfactory performance in terms of over/under shooting and noise amplification.

−0.0311	−0.1016	−0.1478	−0.1016	−0.0311
−0.1016	0.7683	1.0778	0.7683	−0.1016
−0.1478	1.0778	0	1.0778	−0.1478
−0.1016	0.7683	1.0778	0.7683	−0.1016
−0.0311	−0.1016	−0.1478	−0.1016	−0.0311

(a)

-2^{-5}	-2^{-4}	-2^{-3}	-2^{-4}	-2^{-5}
-2^{-4}	2^{-1}	1	2^{-1}	-2^{-4}
-2^{-3}	1	0	1	-2^{-3}
-2^{-4}	2^{-1}	1	2^{-1}	-2^{-4}
-2^{-5}	-2^{-4}	-2^{-3}	-2^{-4}	-2^{-5}

(b)

Figure 5.2. (a) Weights $h[m][n]$ for diffusion approximation, and (b) its approximate implementation using powers of two.

5.2 Sharpening: a contrast maximization problem

The objective to sharpen edges can be expressed as a contrast maximization problem

$$\max_{I} \sum_{s \in I} \sum_{p \in \eta_s} \rho(\nabla I_{p,s}) \tag{5.1}$$

where $\nabla I_{p,s}$ is the gradient between pixel p in spatial neighborhood η_s and center pixel s and $\rho(.)$ is a function designed so that contrast of noisy pixels or those pixels around the strong edges are not enhanced.

It is known that sharpening strong edges creates over/under shooting artifacts around the edges and it is also known that the pixels across strong edges will tend to have large gradient values. Hence, to avoid over/under shooting around edges, these large gradient values should contribute less to the objective function compared to small ones. The distinction between large and medium gradient values can be made using Huber's minmax

norm which is mostly used in robust estimation problems [60]. However, large outliers effects the sensitivity of the estimator. To reduce this sensitivity to large outlier values, Huber's minmax norm switches from L_2 to L_1 norm.

Noise is most visible in smooth regions of the images where the contrast is relatively low [61]. On the other hand, in high-contrast regions, the effect of the noise is generally not noticeable due to the local structural content of the image. Therefore, to avoid boosting the effect of noise in the image, Huber's minmax norm is modified so as to have zero values for the norms of small gradient values. Thus, the modified version of the objective function, $\rho(x)$, technically, will not be a norm since non-zero values are mapped to zero [**?**]. $\rho(x)$ is given by

$$\rho(x) = \begin{cases} 0 & (|x| < T_1) \\ \frac{(|x|-T_1)^2}{2(T_2-T_1)} & (T_1 \leq |x| \leq T_2) \\ |x| - T_2 + \frac{(T_2-T_1)^2}{2(T_2-T_1)} & (T_2 \leq |x|), \end{cases} \tag{5.2}$$

where T_1 and T_2 are threshold values for controlling the noise boosting and over/under shooting protection, respectively. Refer to Figure **??**a for the plot of the function $\rho(x)$.

Maximization problem in (5.1) can be solved with gradient descent as shown in [59]

$$I_s^{(t+1)} = I_s^{(t)} - \frac{\lambda}{|\eta_s|} \sum_{p \in \eta_s} \rho'(\nabla I_{p,s}^{(t)}) \tag{5.3}$$

where $\rho'(x)$ is the first order derivative of $\rho(x)$, λ is step size, and $|\eta_s|$ is the number of pixels in η_s.

As we have discussed before, oscillations can be prevented by stopping backward diffusion after a limited number of iterations. In addition, considering hardware implementation restrictions a filter with compact support is desirable. Therefore, we apply three iterations to prevent oscillations and at the same time keep the hardware requirements low. In three iterations, the diffusion calculations for each pixel involves its neighbor pixels in a 7×7 window when the 8-connectivity neighborhood is used in (5.3). Assuming that $\rho'(x)$ is a linear

function, we will show that diffusion can be approximated with a non-iterative operation. Since the norm of the gradients coming from the boundary pixels on the 7×7 window will be weighted by $(\frac{\lambda}{\eta_s})^3$, which is rather small for typical parameter values, effective window size becomes 5×5 for our non-iterative approximation.

The horizontal and vertical gradients are obtained as

$$\nabla I_{p,s} = I_p - I_s, \quad p \in \eta_s, \tag{5.4}$$

and diagonal gradients as

$$\nabla I_{p,s} = \frac{I_p - I_s}{\sqrt{2}}, \quad p \in \eta_s. \tag{5.5}$$

The diagonal gradients are approximated by normalizing the diagonal intensity differences by $\sqrt{2}$. This prevents the diagonal gradients to dominate the backward diffusion process which in return makes the process more isotropic.

To obtain the non-iterative approximation of the backward diffusion, we assume

$$\rho'(\nabla I_{p,s}) = \rho'(\nabla I_{p,q}) + \rho'(\nabla I_{q,s}), \tag{5.6}$$

where p and q are in 5×5 window and $p, q \neq s$ and s is the center pixel. This assumption is not valid for $\rho(x)$ in (5.2), but is needed to express $I_s^{(3)}$ as the sum of the intensity differences of each pixel with the center pixel. Substituting (5.6) in (5.3) with the gradient approximations as in (5.4) and (5.5), $I_s^{(3)}$ can be represented as a weighted sum of the norms of the intensity differences. Hence the approximate non-iterative solution will be

$$I^{(3)}[m,n] = I[m,n] - \sum_{i,j \in 5\times 5} h[i,j]\rho'(I[m+i, n+j] - I[m,n]) \tag{5.7}$$

Note that a linear low-pass filtering is performed on the modulated intensity differences for pixel $[m,n]$, and the output is added back to the original pixel. For $\lambda = 1$, the filter mask $h[m,n]$ is given in Figure 5.2a which is a 5×5 low-pass filter. A hardware-friendly implementation of Figure 5.2a is given in Figure 5.2b which has coefficients that are powers of two.

5.3 Ringing-aware sharpening

Since sharpening boosts high-frequency components around edges, ringing artifacts are also boosted and becomes easily visible in large flat panel displays. The performance of a sharpening algorithm can be degraded if ringing artifacts are not taken into account in the designing stage. Therefore, we modify our objective maximization problem so that pixels having visible ringing artifacts contribute less to our objective function which we want to maximize. The new maximization problem can be written as

$$\max_{I} \sum_{s \in I} \sum_{p \in \eta_s} \rho^{\dagger}(\nabla I_{p,s}, \pi(s)), \tag{5.8}$$

where $\pi(s)$ is the probability of pixel s to be a ringing-free pixel, and

$$\rho^{\dagger}(\nabla I_{p,s}, \pi(s)) = \rho(\nabla I_{p,s})\pi(s) \tag{5.9}$$

is the new modified objective function for ringing awareness. This formulation implies that if a pixel has a small probability to be a ringing-free pixel than it will be sharpened less. Then the gradient descent solution to this will be

$$I_s^{(t+1)} = I_s^{(t)} - \frac{\lambda}{|n_s|}\pi(s) \sum_{p \in \eta_s} \rho^{'}(\nabla I_{p,s}^{(t)}). \tag{5.10}$$

We again assume linearity given in (5.6) to find an approximate non-iterative solution to the above diffusion process. To obtain a formulation as in (5.7) which is only in terms of neighbor pixel's intensity differences from the center pixel, we further assume $\pi(p) = \pi(s)$ for any $p \in 5 \times 5$ window. The approximate non-iterative ringing-aware solution for pixel s at location $[m, n]$ is

$$I^{(3)}[m, n] = I[m, n] - \pi[m, n] \sum_{i,j \in 5\times 5} h[i, j]\rho^{'}(I[m + i, n + j] - I[m, n]), \tag{5.11}$$

where $h[m, n]$ is as given in Figure 5.2a.

The solution to our maximization problem is independent of how $\pi[m, n]$ is estimated. For a hardware-friendly implementation of the sharpening algorithm, we just need a simple

formulation for $\pi[m,n]$ that can be computed after processing local gradients in a single-pass. This way, once the sum of the weighted gradient norms are obtained in (5.11), $\pi[m,n]$ will be ready to multiply the sum. If we think of gradients coming from the same distribution, pixels of a strong edge in a smooth area can be seen as outlier intensity differences coming from a distribution of small intensity differences. Therefore, the distance of the outlier created by the edge pixel from the estimate of the mean difference is used to obtain $\pi(s)$. We define $\pi(s)$ as below

$$\pi(s) = \frac{1}{T_2}\left(\max_{p\in 5\times 5}\rho^{'}(I_p - I_s) - \frac{\sum_{p\in 5\times 5}\rho^{'}(I_p - I_s)}{25}\right), \tag{5.12}$$

which will be close to one for a a strong edge in a smooth area since the outlier and the mean gradient will be close to T_2 and zero, respectively.

5.4 Experiment Results

We compare our diffusion based sharpening algorithm with the rational UM method [56], and backward diffusion as given in (5.10) using three iterations. Original image given in Figure 5.3a is sharpened using these three algorithms. For a fair comparison we tried to adjust the algorithm parameters so that the vertex of the roof have the same sharpness level in all three sharpened images. Backward diffusion performs better than rational UM in terms of ringing artifact suppression and over/under shooting. In the image sharpened by rational UM, note that the sky region between leaves and the roof suffers more from ringing artifacts. Also, over/under shooting performance can be compared by looking at the areas around the leaves and the roof edges. Our sharpening algorithm given in Figure 5.3d performs very close to the backward diffusion which we are trying to match in terms of sharpening, over/under shooting, noise and ringing amplification performance.

(a) (b)

(c) (d)

Figure 5.3. (a) Original image, (b) enhanced with rational UM, (c) backward diffusion, (d) proposed sharpening.

CHAPTER 6

HYPOTHESIS TESTING FOR PRIOR MOTION INFORMATION

6.1 Introduction

Motion estimation facilitates applications such as motion-compensated noise reduction, frame rate conversion, de-interlacing, and compression [5][54]. Block matching combined with translational-motion model and constancy of brightness assumption is preferred in hardware implementations. In block matching, the motion model applies to all pixels in a block, which simplifies memory access and resource requirements. To improve the performance, efficient buffering algorithms can be designed to fetch a block of pixel data in a small number of clock cycles [62]. However, when there is motion boundaries and/or deformation of objects, block matching can produce large errors in the motion-vector field. Translational-motion model reduces the computation and is sometimes even more robust to noise when compared to more complex models such as the affine motion model [63]. But it fails in the presence of rotation and zooming. After choosing the translational-motion model and its block based region of support, one needs to specify the estimation criteria of the model parameters (*i.e.*, *x* and *y* components of the motion vector). The constancy of brightness assumption tries to minimize the error between a pixel's intensity and its motion-compensated prediction's intensity. In hardware implementations, ℓ_1 norm is used in measuring the error magnitude. Compared to ℓ_2 norm, ℓ_1 norm is more robust in the presence of outliers and saves a multiplication operation [63][60]. ℓ_1 norm accumulated over all pixels in a block is called Sum of Absolute Deviations (SAD). SAD minimization is not sufficient to find true motion vectors, and performs poorly when the brightness in the scene changes.

Clearly, motion estimation is an ill-posed problem, which requires extra information other than pixel intensity data. Spatial and temporal correlation of the motion-vector fields can be used to regularize the motion estimation problem. Spatial correlation is induced

because objects are usually larger than blocks [64]. Moreover, objects usually follow motion trajectories that does not abruptly change, which leads to temporal correlation. The Bayesian framework is promising way to incorporate the prior information. Using Bayes law, a posterior probability for $p(v|d)$ of a realized motion-vector field v is computed by $p(d|v)p(v)$ up to a normalization constant, where $p(d|v)$ is the data likelihood, and $p(v)$ is the prior information.

Markov Random Fields (MRF) is a well-known method to impose spatial correlation. Maximum a posteriori (MAP) estimation of an MRF was introduced into computer vision by Geman and Geman [65]. The MAP-MRF framework can be expressed as an energy-minimization problem. Theoretically, it is possible to find the global minimum using simulated annealing, which is too slow to converge for practical purposes. Recently, approximation algorithms has been designed using graph cuts that iteratively updates the motion field [66][67]. Generally, energy-minimization based motion-estimation algorithms encode only the spatial correlation information via a spatial discontinuity-penalty term in the energy. The computational complexity of an energy-minimization problem that has temporal discontinuity will be too high, especially for hardware implementations: the temporal discontinuity-penalty term using the previous frame would require updating the previous frame's motion-vector field.

Spatial correlation by itself is not sufficient for creating a high-quality motion-vector field. One still needs to utilize the temporal correlation between the previous frame's already computed motion-vector field and the current frame's motion-vector field. By assuming independency, we rewrite $p(v)$ as $p^s(v)p^{\bar{s}}(v)$, where the two terms denote spatial and non-spatial prior motion information. $p^{\bar{s}}(v)$ encodes the temporal correlation between frames, and MAP estimate of $p(v|d)$ can be performed on the current motion-vector field. In addition we show how to use $p^{\bar{s}}(v)$ to pass information from previous resolutions in hierarchical motion estimation. Since, we do not update the previously computed motion-vector field of previous frame or previous resolution, we need to choose which motion vector to

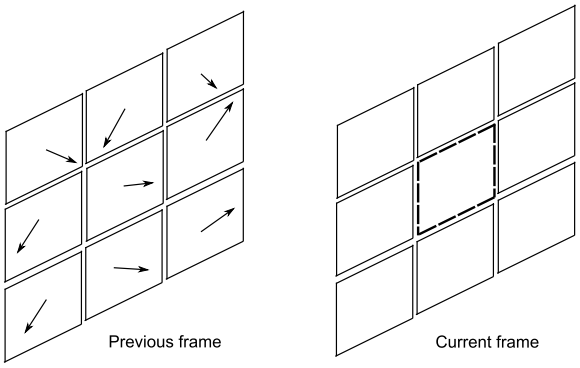


Figure 6.1. Nine motion vectors from the previous frame are hypothesized for the block (with dashed lines) in the current frame)

use in $p^{\bar{s}}(v)$ formulation. We present the prior motion-vector selection as a multi-hypothesis testing problem, and use the *evidence* for the winning hypothesis to adjust the precision[1] of $p^{\bar{s}}(v)$[2].

Next, we will present our Bayesian approach to incorporate non-spatial prior motion information into block matching.

Usually, there are more than one possible prior motion vectors (v_i^p) available (*e.g.*, see Figure 6.1)[3]. Unfortunately, we do not know which block's motion vector in the previous frame applies to the current block, because this is actually the motion estimation problem we are trying to solve. But we can expect that the current block must be a displaced version of one of the blocks that are not too far from its location in the previous frame. To choose the best v_i^p from a set of motion vectors, $\{v_i^p\}$, we use multiple hypothesis testing, which is described next.

6.2 Multiple Hypothesis Testing for Prior Selection

We have a set of hypotheses $\{H_1, H_2, \ldots, H_N\}$ to be tested. H_i hypothesizes that the motion vector v is equal to v_i^p from the previous frame. We want to find the hypothesis that has the

[1] We refer to reliability of the prior as precision, following the convention for Gaussian distribution, which we use to model the prior distribution.

[2] From now on, we will drop the superscript $\bar{s}$ for notational simplicity.

[3] Superscript p denotes data from the previous frame.

highest *evidence*. The evidence value for H_i is defined as

$$e(H_i) = \ln \frac{p(H_i|d, v^p)}{p(\overline{H}_i|d, v^p)}, \tag{6.1}$$

where d denotes data (pixel intensities), v^p denotes prior information on previous frame's motion vectors, and $\overline{H}_i$ implies H_i is false [68]. $p(H_i|d, v^p)$ and $p(\overline{H}_i|d, v^p)$ are posterior probabilities of H_i and $\overline{H}_i$ obtained via Bayes Law:

$$p(H_i|d, v^p) = p(H_i|v^p)\frac{p(d|H_i, v^p)}{p(d|v^p)}, \tag{6.2}$$

$$p(\overline{H}_i|d, v^p) = p(\overline{H}_i|v^p)\frac{p(d|\overline{H}_i, v^p)}{p(d|v^p)}, \tag{6.3}$$

where $p(H_i|v^p)$ and $p(\overline{H}_i|v^p)$ denote prior information on H_i and $\overline{H}_i$, respectively.

Using (6.2) and (6.3) in (6.1)

$$e(H_i) = \ln \frac{p(H_i|v^p)p(d|H_i, v^p)}{p(\overline{H}_i|v^p)p(d|\overline{H}_i, v^p)} \tag{6.4}$$

By applications of Bayes Law on $p(d|\overline{H}_i, v^p)$, $e(H_i)$ becomes

$$e(H_i) = \ln \frac{p(d|H_i, v^p)p(H_i|v^p)}{\sum_{k=1, s.t. k\neq i}^{N} p(d|H_k, v^p)p(H_k|v^p)}. \tag{6.5}$$

In the above formula, $p(H_i|v^p)$ represents our prior information on H_i before observing any data (*i.e.* the current frame). Generally, motion estimation algorithms output a motion vector and an associated confidence value c_i^p for that motion vector. Hence, before processing a new frame if we know that v_i^p has a high confidence, then it is more likely to be an estimate of a true motion in the image. This makes it more likely to survive in the new frame. Therefore, we can compute $p(H_i|v^p)$ by

$$p(H_i|v_p) = \frac{c_i^p}{\sum_{k=1}^{N} c_k^p}. \tag{6.6}$$

To select the best prior, we solve

$$i^* = \arg \max_{i\in\{1,2,\ldots,N\}} e(H_i). \tag{6.7}$$

However, if we do not use any prior information on H_i's by assuming $p(H_i|v^p)$ is uniform, (6.5) is simplified to

$$e(H_i) = \ln \frac{p(d|H_i, v^p)}{\sum_{k=1,}^{N} p(d|H_k, v^p) - p(d|H_i, v^p)}. \tag{6.8}$$

Since $e(H_i)$ in (6.8) is a strictly increasing function of $p(d|H_i, v^p)$, maximization problem in (6.7) becomes

$$i^* = \arg \max_{i \in \{1,2,\ldots,N\}} p(d|H_i, v^p), \tag{6.9}$$

which is simpler and does not involve division as in (6.7).

Although solving the simpler maximization in (6.9) gives the optimal i^* for the maximization in (6.7) by assuming uniform prior on H_i's, we still need to compute $e(H_i)$. This is because we use the top two largest evidence values, $e(H_{i^*})$ and $e(H_{i^{**}})$ to adjust the precision, $\frac{1}{\sigma^2}$, of $p(v)$ as below

$$\frac{1}{\sigma^2} = f(e(H_{i^*}) - e(H_{i^{**}})), \tag{6.10}$$

where f is a non-decreasing, non-negative function. A large difference between the two largest evidence values implies that the two hypothesis are well separated and we can be more certain that our decision of selecting H_{i^*} is right.

We would like to simplify $e(H_i)$ further by an approximation. When the evidence is large for H_{i^*} (*i.e.*, $p(d|H_{i^*}, v^p)$ is large), it should be saturated to avoid assigning too high $\frac{1}{\sigma^2}$. When the evidence is small, we want to adjust $\frac{1}{\sigma^2}$ by (6.10). Hence, our approximation of $e(H_i)$ should especially work well for small $p(d|H_i, v^p)$. For $\sum_{k=1,}^{N} p(d|H_k, v^p) \gg p(d|H_i, v^p)$, we can approximate $e(H_i)$ by

$$e(H_i) = \ln \frac{p(d|H_i, v^p)}{\sum_{k=1,}^{N} p(d|H_k, v^p)}. \tag{6.11}$$

Substituting (6.11) in (6.10), we get

$$\frac{1}{\sigma^2} = f(\ln p(d|H_{i^*}, v^p) - \ln p(d|H_{i^{**}}, v^p)), \tag{6.12}$$

which is the difference of the likelihoods of the top-two hypothesis.

Above formula for $\frac{1}{\sigma^2}$ measures if H_{i^*} is well separated from the rest of the hypothesis or not. However, the way we set our hypotheses can reduce the inference of a good H_{i^*}. Our H_i's are *simple* hypotheses meaning that each hypothesis specifies a single value to v (*i.e.*, $H_i : v = v_i^p$). This means that close but not identical (*e.g.*, sub-pixel different) v_i^p's all of which applies well to the current frame, can be assigned to different H_i's. But from (6.11), $\frac{1}{\sigma^2}$ will be small since both H_{i^*} and $H_{i^{**}}$ will have high likelihoods. To overcome this, one can impose a minimum distance among the hypothesized v_i^p's.

One last improvement on our hypotheses set is to include a *dummy* hypothesis, H^D, to represent cases like occlusion and scene change in which no prior motion information is available for the next frame. Adding a *dummy* hypothesis enables us to choose H^D when evidence for v_i^p's are small. Obviously, the likelihood given H^D does not depend on pixel intensity differences, hence we set $p(d|H^D, v^p) = \epsilon$, where ϵ is a small number. When H^D is selected, we do not have any informative prior information $p(v)$ to update $p(d|v)$ because there is no match of pixels by the definition of H^D. Therefore, we need to set $\frac{1}{\sigma^2}$ to zero. To do this by smoothly changing $\frac{1}{\sigma^2}$ as H^D becomes more likely to be accepted, we can modify (6.10) to

$$\frac{1}{\sigma^2} = f[(\ln p(d|H_{i^*}, v^p) - \ln p(d|H_{i^{**}}, v^p))(\ln p(d|H_{i^*}, v^p) - \ln \epsilon)]. \tag{6.13}$$

As the likelihood $p(d|H_{i^*}, v^p)$ of the selected hypothesis H_{i^*} decreases (H^D becomes more plausible), the precision $\frac{1}{\sigma^2}$ of the prior information also decreases and becomes exactly zero when H^D is selected.

6.3 Computing the Posterior Distribution

After selecting the prior motion vector $v_{i^*}^p$ and computing its precision $\frac{1}{\sigma^2}$, we need to model the prior distribution $p(v)$. We expect the true motion vector to be *closely* distributed around $v_{i^*}^p$, and $\frac{1}{\sigma^2}$ hints how close this is. In a sense, $v_{i^*}^p$ and $\frac{1}{\sigma^2}$ specifies the first and the second moments of the prior distribution. To avoid imposing any further constraints on the prior

distribution, Gaussian distribution is chosen to model $p(v)$ according to the principle of maximum entropy [68]

$$p(v) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(v-v_{i*}^p)'(v-v_{i*}^p)}, \tag{6.14}$$

which assumes horizontal and vertical deviations from v_i^p are not correlated.

The error between a pixel x's intensity and its reference pixel's intensity obtained by motion compensating with v is denoted by d_v^x, and it is created by many error sources such as noise, aliasing, compression artifacts, deformation, zooming, rotation, brightness change, etc. Again, by the principle of maximum entropy to account for all these sources Gaussian distribution is used for modeling data likelihood

$$p(d_v^x|v) = \frac{1}{\sqrt{2\pi}\tau} e^{-\frac{1}{2\tau^2}d_v^{x2}}, \tag{6.15}$$

where τ is a constant variance term reflecting the strength of the above mentioned error sources in the video sequence. Since block matching assumes the same motion vector v applies to all pixels in block B, the data likelihood for B is

$$p(d|v) = \prod_{x\in B} p(d_v^x|v), \tag{6.16}$$

$$= \frac{1}{\sqrt{2\pi}\tau} e^{-\frac{1}{2\tau^2}\sum_{x\in B} d_v^{x2}}, \tag{6.17}$$

by assuming d_v^x's are independently distributed.

From (6.17) and (6.14), the posterior distribution of v is

$$\begin{aligned} p(v|d) &\propto p(d|v)p(v) \\ &\propto \frac{1}{2\pi\sigma\tau} e^{-\frac{1}{2\tau^2}\sum_{x\in B} d_v^{x2}} e^{-\frac{1}{2\sigma^2}(v-v_{i*}^p)'(v-v_{i*}^p)} \end{aligned} \tag{6.18}$$

The log of the posterior is

$$\log p(v|d) \propto -\frac{1}{2\tau^2}\sum_{x\in B} d_v^{x2} - \frac{1}{2\sigma^2}(v-v_{i*}^p)'(v-v_{*i}^p) \tag{6.19}$$

MAP estimate, v^{MAP}, is found by minimizing $-\log p(v|d)$ given by

$$v^{MAP} = \arg\min_v \frac{1}{2\tau^2}\sum_{x\in B} d_v^{x2} + \frac{1}{2\sigma^2}(v-v_{i*}^p)'(v-v_{*i}^p). \tag{6.20}$$

Substituting (6.17) in (6.13), $\frac{1}{\sigma^2}$ can also be simplified to

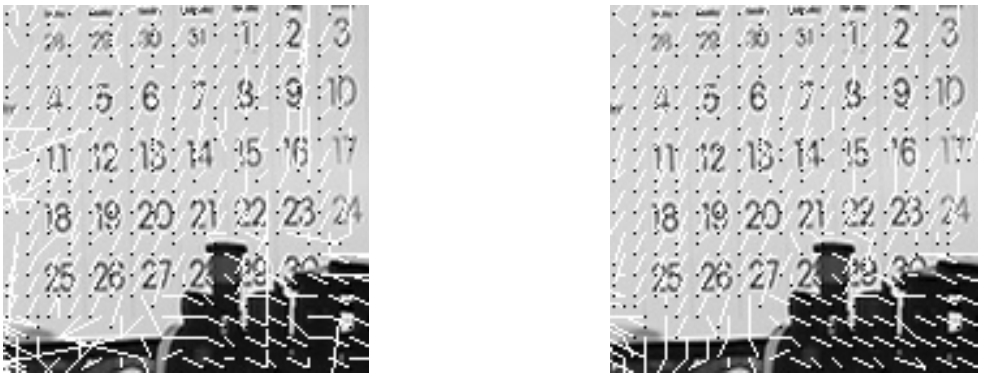
$$\frac{1}{\sigma^2} = f[(\frac{1}{2\tau^2}\sum_{x\in B} {d^x_{v^p_{i^{**}}}}^2 - \frac{1}{2\tau^2}\sum_{x\in B} {d^x_{v^p_{i^*}}}^2)(\ln\frac{1}{\epsilon} - \frac{1}{2\tau^2}\sum_{x\in B} {d^x_{v^p_{i^*}}}^2)]. \tag{6.21}$$

Inspecting (6.20), we can see that the MAP estimate minimizes a bi-criterion cost function. The first term is the data term, which is a sum of square errors (SSE). The second term penalizes deviations from the hypothesized motion vector. The weight of the second term is adjusted by how well the hypothesized motion vector applies to the current block B, which is inferred using (6.21). If the data likelihood and prior distributions are modeled with Laplace distribution instead of a Gaussian distribution, all the SSE terms in (6.20) and (6.21) would be SAD terms, which is more hardware friendly. Searching for the minimum of the first term alone corresponds to regular block matching that minimizes SAD.

The search window, S, is exhaustively searched for finding v^{MAP}. Heuristics search algorithms such as three-step search [69], cross-search [70] for minimizing (6.20) can also be used instead of an exhaustive search. Furthermore, although the first term is a non-convex cost function, the second term is a convex cost function, which becomes more dominant as v's distant from $v^p_{i^*}$ are searched. This can be taken advantage of, while designing heuristic search algorithms specifically for minimizing (6.20). More costly searches such as full-search can be performed when non-convex cost function is dominant (*i.e.*, v's close to $v^p_{i^*}$) to avoid getting stuck at a local minimum, and heuristic search patterns can be utilized otherwise.

6.4 Choosing The Best Set of Hypotheses

To incorporate prior motion information from the previous frame, the hypothesis set should cover a large enough area in the previous frame that contains B's pixels while keeping the computation at an acceptable level. Obviously, this depends on the motion type. As shown in Figure 6.1, in addition to B's collocated block in the previous frame, its 3×3 block neighborhood can also be used. With the *dummy* hypothesis, this will make a total of ten hypotheses to utilize the available prior information from the previous frame. In



(a) (b)

Figure 6.2. Block matching results for a two-resolution hierarchical motion estimation: (a) SAD minimization (b) minimization of bi-criterion cost in Equation (6.20).

hierarchical motion estimation, the hypothesis set needs to be designed differently. For example, if down-sampling by two is performed to create the coarser resolution frame from the finer resolution frame, B in the finer resolution will effectively correspond to a quarter-sized block in the coarser resolution. Hence, its motion information may be lost if there is a motion boundary in the whole block. To include this lost information in our hypothesis set, we need to use other neighbor blocks in the coarse resolution. For any B at location (r, c), corresponding to row and column number respectively, we need to use four blocks in the coarse resolution to account for all possible motion boundary directions. These four blocks are located at $(\lfloor \frac{r+1}{2} \rfloor, \lfloor \frac{c+1}{2} \rfloor)$, $(\lfloor \frac{r+1}{2} - 1 \rfloor, \lfloor \frac{c+1}{2} \rfloor)$, $(\lfloor \frac{r+1}{2} \rfloor, \lfloor \frac{c+1}{2} - 1 \rfloor)$, $(\lfloor \frac{r+1}{2} - 1 \rfloor, \lfloor \frac{c+1}{2} - 1 \rfloor)$. With the *dummy* hypothesis, there will be five hypotheses in our set.

6.5 Experiment Results

We tested our proposed method using standard video test sequences. In all the test sequences, using prior motion information has improved the quality of the motion-vector field. Due to space limitations, we present some visual results from the Mobile and Calendar sequence. In the following figures, motion vector of a block is represented with a white line and its direction is denoted by a block dot. Unfortunately, we can not do any mean square error (MSE) comparison by measuring the error between the original image

(a) (b)

Figure 6.3. Block matching results for passing prior motion information from the previous frame: (a) SAD minimization (b) minimization of bi-criterion cost in Equation (6.20).

and the reconstructed image via motion compensation. It is obvious from (6.20) that SAD minimization will always give a better MSE because of our second cost term. Therefore, we present visual results in which motion vectors are amplified by two.

Our first result demonstrates the improvement provided by passing prior motion information from a hierarchical motion estimation with two resolution levels. Figure 6.2(a), and 6.2(b) shows the motion-vector field from the 25^{th} frame, produced by SAD minimization and our proposed bi-criterion cost minimization in (6.20). The hypothesis set is as described in Section 6.4. Due to noise in the image, there are some random motion vectors in Figure 6.2(a), which are worst in the smooth areas on the calendar. The motion-vector field in Figure 6.2(b) looks more consistent, most of the random vectors are corrected with the help of the prior information passed from the previous resolution.

The next result is produced by passing prior motion information from the previous frame. Figure 6.3(a) and Figure 6.3(b), shows the motion-vector field from the 22^{nd} frame produced by SAD minimization and our proposed method, respectively. Using the previous frame our proposed method improves the performance of block matching for repeating structures (*i.e.*, spiral calendar perforations).

It is important to note that, the quality of the motion-vector field still needs to be improved. By iteratively minimizing an energy function designed to incorporate spatial smoothness constraint into gray-level matching will improve the estimated motion-vector field, which is the subject of the next chapter.

CHAPTER 7

ENERGY MINIMIZATION BASED MOTION ESTIMATION

7.1 Introduction

Motion estimation is an ill-posed problem because images are projections of 3D scenes onto 2D planes and large amount of information is lost as a result. Matching gray-levels of image pixels only utilizes the brightness constancy assumption, which performs poorly for motion estimation. Oftentimes, more than one motion-vector can be good solutions to gray-level matching, because of noise, compression artifacts, lack of detail, repeating structures, etc. To favor the true solution against others, one needs to use extra information. Spatial correlation in motion-vector fields is one such source of information in addition to the non-spatial priors discussed in Chapter 6. One can design an objective function that incorporates spatial priors into the motion estimation problem.

An objective function measures how well a particular solution from the large solution space fits both to observed data and prior knowledge. To exploit spatial smoothness in motion estimation an objective function should contain a data and a smoothness term. The data term determines if a solution explains the observed data by using the constancy of brightness assumption (*i.e.* gray-level matching). The smoothness term measures variations in the motion-vector field. Both terms are designed so that small values are desired. Objective functions are often called energy functions because of minimization algorithms such as Simulated Annealing (SA), which is inspired from annealing in metallurgy; a technique that involves controlled cooling of a material [71].

Since a naive search for global optimal solution is exponentially complex, the energy function is minimized using a local search algorithm, which starts with an initial solution and searches for better solutions using neighborhoods. The energy has many local minima, which can cause the minimization algorithm perform poorly. For example, an image of width W, height H, block size BS, and number of possible vectors, S, the number of

solutions depends exponentially on S and is about $(S)^{2\frac{W}{BS}\frac{H}{BS}}$. To prevent stopping at a bad local minimum and to enable convergence to a good one among possible local minima requires complex algorithms, hence, energy-minimization for motion estimation is a highly complex combinatorial optimization problem. Finding the global minimum is often NP-hard, hence the problem necessitates local search algorithms that are fast converging with low computational complexity for hardware friendliness.

Energy minimization is widely used in motion estimation. Horn and Schunck was first to include the smoothness term into the energy function for optical flow estimation[72]. Boykov et al. [73] and Kolmogorov et al. [66] presented graph-cut based approximate solutions to find disparity maps between stereo images.

7.1.1 When does energy minimization fail?

In case the energy-minimization algorithm results in a bad solution, it is hard to determine the cause of failure. The failure may be due to a poorly designed energy function or an approximation algorithm that stopped at a local minimum not close to the global minimum.

The reasons for a poorly designed energy function are related to the smoothness term. Smoothness term measures the amount of variation in the motion-vector field by considering motion vectors in a local neighborhood. Energy-minimization algorithms try to minimize the data term and smoothness terms at the same time. By favoring towards a smoothly varying motion-vector field at the cost of a greater gray-level mismatch, the prior knowledge that pixels on the same object tend to move together is imposed. However, this is not valid for object boundaries, deformations, and rotation and zooming under the translational motion model. Paradoxically, motion segmentation of the image must be known to correctly impose the smoothness prior. Motion segmentation will determine which pixels/blocks in the local neighborhood are needed to be used to impose smoothness.

A bad solution can as well result from the failure of the energy-minimization algorithm (*i.e.*, convergence to a poor local minimum). Local search algorithms reduce the energy at each time, hence it is not possible to break away from the local minimum once attained.

One way to overcome this is by jumping to a distant point in the solution space. However, this requires changing multiple motion vectors at each step, which increases the complexity (*e.g.*, if two blocks' motion vectors can change rather than one block's, the number of changes over i steps increases from iS to iS^2). This requires more computation and also memory resources.

Before discussing the details of our approach, some clarification is needed for the terminology we are using: A move is a change of motion vectors for reducing the energy. A standard-move is a change of a single block's motion vector. An iteration is a combination of moves, possibly processing all blocks/pixels for their computing their best moves. Each iteration brings the solution closer to a local minimum of the energy.

7.1.2 A joint approach: The primal-dual method

Our contribution to energy-minimization based motion estimation is in solving the two aforementioned problems: energy-function design and fast convergence of the energy-minimization algorithm to a *good* minimum. Contrary to previous works, our approach is a joint approach in which the energy function is not fixed at the beginning but updated as the energy-minimization algorithm iterates towards a minimum. Prior works on energy-minimization determines the energy function at the beginning by fixing the parameters manually, and do not modify the parameters as the solution is improved. Energy-minimization is commonly used in computer vision applications [74][75], and it was applied to pixel-level stereo-disparity estimation [66][67][76], which is similar to motion estimation in essence. In this work we use energy-minimization for block-based motion estimation.

Our energy function consists of a data term that relies on the brightness consistency assumption applied to a block of pixels and a smoothness term that measures thresholded first-order variations in the motion-vector field. Thresholding in smoothness computation preserves discontinuities in the motion-vector field[76], which affects the performance

around motion boundaries. A good objective function for motion estimation correctly imposes the smoothness constraint, which requires the motion segmentation of the image. This is clearly a chicken and egg dilemma. We overcome this causality dilemma between motion estimation and motion segmentation by correcting our energy function as we learn more about the motion-vector field throughout the iterations. We show that the dual problem of the primal problem (*i.e.*, energy minimization) for motion estimation can be used to infer about the motion segmentation. Hence as the iterations progress, primal-dual relation provides us information on how to redesign the energy function. This way, the energy function better imposes the smoothness constraint.

Even if the perfect energy function is designed, the energy-minimization algorithm can still converge to an undesired local minimum. Many local minima are possible since the solution space is large and the data term in the energy function is non-convex because of independent motion of objects, and the aforementioned reasons for insufficient gray-level information. Since, we want to design a standard-move algorithm for hardware friendliness, it is difficult to break away from a local minimum once arrived. We try to overcome this problem by increasing the weight of the smoothness term which is convex. By the use of the primal-dual relation we can impose a better smoothness constraint since it is corrected throughout the iterations. This enables us to increase the weight of the smoothness term in a new iteration compared to the previous iteration. Increasing the smoothness term's weight, effectively implies less weight for the data term and decreases the chance of getting stuck at a local minimum due to the non-convexities of the data term.

In block-based motion estimation, a motion vector v_b is computed for each block $b \in B$, where B is the set of all blocks in an image. Motion estimation can be formulated as an energy-minimization problem with an energy function

$$E(v) = \sum_{b \in B} D_b(v_b) + \lambda \sum_{\{b,a\} \in \mathcal{N}} V_{b,a}(v_b, v_a), \tag{7.1}$$

where the first term measures how well b's motion vector v_b explains the observed data, and the second term measures the smoothness of the motion-vector field. The overall weight

of the prior smoothness information is given by λ. $\mathcal{N}$ denotes block pairs that interact with each other, which is often defined as adjacent blocks (*e.g.*, b interacts with its eight-connectivity neighborhood). The penalty function $V_{b,a}$ depends on the block pairs and determines the penalty for variations in v.

Using the notation in the previous chapter, the data term is given as

$$D_b(v_b) = \sum_{x \in b} {d_{v_b}^x}^2, \tag{7.2}$$

where $d_{v_b}^x$ is the gray-level error between pixel $x \in b$ and its motion compensated version using v_b in the reference image. A true motion vector will produce a small D_b by the brightness-constancy assumption.

7.1.3 Energy function as the log of the posterior

From (7.1), one can see that the contribution of b to the total energy is independent of the other blocks given the pairs it interacts with. This is a Markovian property which paves the way for a probabilistic justification of the energy function as a posterior probability. Using the equivalence between MRFs and Gibbs random fields [77], we can use the Gibbs probability distribution and an MRF with cliques of size two to obtain

$$P(v) = Z^{-1} exp\left(-\lambda \sum_{\{b,a\} \in \mathcal{N}} V_{b,a}(v_b, v_a)\right), \tag{7.3}$$

where Z is a normalizing constant. Using the data likelihood as in (6.17) for all $b \in B$

$$P(d|v) \propto exp\left(-\sum_{b \in B} D_b(v_b)\right) \tag{7.4}$$

The posterior is given by the Bayes rule

$$p(v|d) \propto exp\left(-\sum_{b \in B} D_b(v_b) - \lambda \sum_{\{b,a\} \in \mathcal{N}} V_{b,a}(v_b, v_a)\right) \tag{7.5}$$

The log of the posterior is the negative of the energy function

$$\log(p(v|d)) \propto -\left(\sum_{b \in B} D_b(v_b) + \lambda \sum_{\{b,a\} \in \mathcal{N}} V_{b,a}(v_b, v_a)\right). \tag{7.6}$$

Hence, the MAP estimation is equivalent to minimizing (7.1).

7.1.4 Types of the smoothness prior

There are three major types of smoothness terms. The first type is usually called as the Potts model

$$V_{b,a}(v_b, v_a) = \alpha I(v_b \neq v_a), \tag{7.7}$$

where I is the indicator function. The Potts model assigns the same penalty to all discontinuities and therefore is discontinuity-preserving. It is suitable for problems in which the solution has discontinuities such as image restoration with binary images. The second type is the linear smoothness model

$$V_{b,a}(v_b, v_a) = \alpha|v_b - v_a|, \tag{7.8}$$

which assigns larger penalty when two adjacent blocks move in farther directions. However, this smoothes the motion-vector field at the object boundaries. To overcome this, a truncated linear model can be used

$$V_{b,a}(v_b, v_a) = \min(K, \alpha|v_b - v_a|), \tag{7.9}$$

where K is a constant. When the difference of the motion vectors is greater than K under L_1 norm, the penalty is truncated to K and therefore does not distinguish between greater discontinuities. This helps with preserving the motion boundaries compared to the linear model. Another good property of the truncated linear model as well as the linear model is that, they are both metrics on v. This is a required property for the graph-cut based α-expansion algorithm that requires $V_{b,a}$ to be a metric [73].

7.1.5 Breaking away from *bad* local minima

A block-based motion-vector field v is a local minimum of the energy E if

$$E(v) < E(v') \quad \forall v' \quad ''near\ to'' \quad v, \tag{7.10}$$

where v' is near to v means that v' is within a single move of v. A motion-vector field $v' \neq v$ is a single move away from v if it can be reached in one step. Various types of

moves can be designed. The simplest one is a standard move, which is a move that can only change one block's motion vector at a step. Algorithms based on standard moves is prone to getting stuck at local minima, because they can not jump to farther locations in the solution space. Especially when all or most of the blocks interacting with a block have *bad* motion vectors, a standard move will not likely be able to correct the motion vectors over many iterations because of the incorrect smoothness constraint that will be imposed by its neighbors. One way to deal with this is to allow large moves at each iteration, in which a group of blocks can change their motion vectors together. Another way would be to start the iterations with a *good* initial motion-vector field v that has few such local minima. The initial motion-vector field can be obtained by using non-spatial prior information from the previous frame or the previous resolution in a hierarchical motion estimation. As shown in Chapter 6, hypothesis testing can be used to choose which prior motion vector to use and its contribution in obtaining the initial v. We choose utilizing non-spatial priors combined with standard-move iterations to avoid computation and memory requirements of large-move algorithms. This cuts down the number of *bad* local minima and hence increases the likelihood of converging to a good solution. In addition to this, standard-moves are hardware-friendly and easy to implement because they do not need information from other blocks (*i.e.* Each block makes its own standard-move decision). They require less memory, computation, and enable parallelization of the algorithm[1]

We will next present how to efficiently make large moves using graph-cuts to discuss their advantages in solving local minima issues and disadvantages in terms of computational complexity. Then we will present our proposed standard-move algorithm.

[1]For example, a parallelized implementation of our proposed solution using General Purpose Graphic Processing Units (GPGPUs) requires dividing image blocks into independent groups of blocks (GOBs). These GOBs can than be assigned to different processors and start running independent of each other. Our proposed algorithm given at the end if this chapter, typically gives a 10x to 100x improvement for an SD image over CPU's average running times (close to one second per frame using Intel's dual core CPU) depending on the GPGPU's number of cores. This is fast enough to make the implementation work in real time.

7.2 A large-move algorithm based on graph-cuts

As mentioned earlier, the number of possible moves grows exponentially when large moves are allowed (*e.g.*, if all blocks' vectors can be changed, than the number of possible moves increases from $|S||B|$ to $|S|^{|B|}$). Naively checking all possible moves requires an exponential complexity, hence low-complexity approximation algorithms are required to find a *good* move. The α-expansion algorithm is a greedy algorithm that uses graph-cuts to make large moves, and has been recently used in computer vision problems [73][76].

The α-expansion algorithm executes *cycles* until convergence and each cycle iterates over all the motion vectors in S (called *labels*) and tries to reduce the energy. Following the definition given in (7.10), the convergence to a local minimum is reached when an iteration does not decrease the energy. In an iteration for label α, a move from v to v' is reached by changing any set of block's motion vector to α and keeping all other blocks' motion vectors. In other words, α-expansion tries to expand the regions in an image that belong to a particular motion vector at each iteration.

Graph-cuts are utilized in α-expansion iterations. It has been shown in [73] that an α-expansion move to v' corresponds to a cut on a specially constructed graph G using v obtained from the previous iteration. G has two terminals α and $\overline{\alpha}$, and the blocks in the partition containing α change their labels to α and the blocks in the other partition keep their labels. An α-expansion move and a cut have the same energy that differs by a constant that is independent of the cuts [73]. Hence, the minimum cut in G will find the best greedy move among all possible α-expansions.

The minimum cut is found using the maximum flow between α and $\overline{\alpha}$ according to Max-Flow-Min-Cut theorem of Ford and Fulkerson [78]. The Ford-Fulkerson algorithm tries to find the maximum flow from terminal α to terminal $\overline{\alpha}$ by pushing flow on paths between the two terminals without violating the edge capacities in the graph. Edmonds-Karp algorithm is a specialization of the Ford-Fulkerson algorithm that finds paths via

breadth-first search[2][79]. Edmonds-Karp algorithm has a lower complexity $O(8|B|^3)$. A more recent work on computing the maximum flow is [67]. It works better on typical computer vision problems but has a higher worst-case complexity, which determines the complexity of a hardware implementation.

Although large moves converge faster to better local minima and are significantly faster than exhaustive searches, the computational complexity renders them useless for hardware implementations. For example, Edmonds-Karp algorithm for a single α-expansion will have complexity $O(8|B|^3)$ using eight-connectivity neighborhood, and iterating through all the motion vectors (labels) during a cycle will increase the complexity to $O(8|S||B|^3)$, which is still much faster than an exhaustive search $O(|S|^{|B|})$. Since the complexity is proportional to $|B|^3$, which will increase the complexity exponentially when blocks are partitioned into smaller blocks to improve the quality of v. Fortunately, standard-move algorithms' complexity is $O(|S||B|)$, which depends linearly on $|B|$. We will next present our single-move algorithm, discuss its convergence rate and worst-case energy at convergence.

7.3 A single-move algorithm

A single-move can only change a single block q's motion vector v_q. To find the best move, we need to search for a v_q that minimizes the energy. Let us rewrite (7.1) by separating terms that do not involve q from the rest

$$E(v) = D_q(v_q) + \lambda \sum_{\{b,a\}\in\mathcal{N}_q} V_{b,a}(v_b, v_a) + \sum_{b\in B, b\neq q} D_b(v_b) + \lambda \sum_{\{b,a\}\in\mathcal{N}-\mathcal{N}_q,} V_{b,a}(v_b, v_a), \qquad (7.11)$$

where $\mathcal{N}_q$ denotes block pairs interacting with q, which are the blocks in its eight-connectivity neighborhood. Since the last two terms do not depend on q, they can be ignored. Hence, the best standard-move for q is given by

$$v_q^* = \arg\min_{v_q\in S} D_q(v_q) + \lambda \sum_{\{b,a\}\in\mathcal{N}_q} V_{b,a}(v_b, v_a). \qquad (7.12)$$

[2] A breadth-first search in graph theory begins at the root node and then searches all the neighboring nodes. Then for each searched node, it searches their un-searched neighbor nodes.

Vectors that try to satisfy the brightness constancy assumption for pixels in q, while agreeing with other vectors in N_q will more likely be chosen, as dictated by the above equation for standard-moves.

7.3.1 Upper bound on the energy gap

We prove that a local minimum under standard-moves obtained by (7.12) is bounded with global minimum plus a term that depends on λ and the penalty function V.

Let

$$M = \max V_{b,a}(v_b, v_a), \quad v_b, v_a \in S \tag{7.13}$$

be the maximum penalty value between any two interacting blocks. Let $\overline{v}$ be a local minimum under standard-moves, and $\overline{\overline{v}}$ be the globally optimal solution.

Theorem 7.3.1 *The energy gap, $E(\overline{v}) - E(\overline{\overline{v}})$, between a local minimum and the global minimum is upper-bounded by λM.*

Proof Since $\overline{v}$ is a local minimum, by (7.12) we know that for any block b

$$D_b(\overline{v}_b) + \lambda \sum_{\{b,a\} \in \mathcal{N}_b} V_{b,a}(\overline{v}_b, \overline{v}_a) \leq D_b(\overline{\overline{v}}_b) + \lambda \sum_{\{b,a\} \in \mathcal{N}_b} V_{b,a}(\overline{\overline{v}}_b, \overline{\overline{v}}_a). \tag{7.14}$$

To obtain the total energy of $\overline{v}$ and $\overline{\overline{v}}$, we need to sum (7.14) over all blocks in B:

$$E(\overline{v}) + \lambda \sum_{\{b,a\} \in \mathcal{N}} V_{b,a}(\overline{v}_b, \overline{v}_a) \leq E(\overline{\overline{v}}) + \lambda \sum_{\{b,a\} \in \mathcal{N}} V_{b,a}(\overline{\overline{v}}_b, \overline{\overline{v}}_a) \tag{7.15}$$

which is obtained by using the energy definition in (7.1) and observing that by summing (7.14) over all blocks, the penalty terms between interacting blocks are summed twice (*e.g.*, $\{b, a\} \in \mathcal{N}_b$ and $\{b, a\} \in \mathcal{N}_a$). By rearranging terms in (7.15)

$$E(\overline{v}) - E(\overline{\overline{v}}) \leq \lambda \sum_{\{b,a\} \in \mathcal{N}} V_{b,a}(\overline{\overline{v}}_b, \overline{\overline{v}}_a) - \lambda \sum_{\{b,a\} \in \mathcal{N}} V_{b,a}(\overline{v}_b, \overline{v}_a), \tag{7.16}$$

Since $V_{b,a}(v_b, v_a) \geq 0$ and $V_{b,a}(v_b, v_a) \leq M$ (from (7.13)), we can simplify (7.16) to

$$E(\overline{v}) - E(\overline{\overline{v}}) \leq \lambda M. \tag{7.17}$$

□

Hence the energy gap $E(\overline{v}) - E(\overline{\overline{v}})$ is bounded by λM. When λ is set to zero, $E(\overline{v}) = E(\overline{\overline{v}})$. Since there will be no interaction between blocks for $\lambda = 0$, for each block the vector that minimizes $D_b(v_b)$ will be chosen by the standard-move algorithm, which is also the globally optimal solution.

To decrease the energy gap, λM must be decreased. However, decreasing λM also affects the quality of $\overline{\overline{v}}$ since it changes the energy function, which measures the quality. In other words, although having a tighter bound means that $E(\overline{v})$ will be closer to $E(\overline{\overline{v}})$, it does not necessarily mean that $\overline{v}$ is better, because $\overline{\overline{v}}$ that we are trying to approximate can be inferior. We need to find a way to decrease λM without sacrificing the quality of $\overline{\overline{v}}$. Reducing λM can be achieved by either reducing λ or M. λ determines how much the prior smoothness information is reliable compared to brightness constancy assumption, hence its design must also take into account the level of noise and compression artifacts. Although setting λ to zero will make the energy gap zero, it is not desired since we will not be using any prior information about v's smoothness. Reducing M, (*i.e.*, the maximum value of the penalty term) will also tighten the upper bound for local minimum's energy and it may even improve the quality of $\overline{\overline{v}}$ because truncated linear models preserve motion boundaries from smoothing. They have smaller M values, which means a lower energy gap. Substituting (7.9) and (7.13) in (7.17), the upper bound is given by

$$E(\overline{v}) \leq E(\overline{\overline{v}}) + \lambda K, \tag{7.18}$$

where $K < M$. This shows that using truncated linear models not only improve the quality of $\overline{\overline{v}}$ by preserving motion boundaries, but also improves the energy gap of standard-move algorithms. Hence, the value of K in a truncated linear model is crucially important in the quality of the computed v.

7.3.2 Lower bound on the convergence time

Another important aspect of our standard-move algorithm's quality[3] is its convergence time. Fast-converging algorithms make pipelining design easier and saves resources in hardware implementations. To obtain a lower bound on convergence time, we first need to find the minimum amount of decrease in energy at each iteration. Let

$$m = \min_{v_b \neq v_a} V_{b,a}(v_b, v_a), \quad v_b, v_a \in S \tag{7.19}$$

be the minimum penalty between interacting pairs and let v^o be the initial v. Then the minimum amount of decrease in energy per iteration is λm. In the worst case, the convergence to $\overline{v}$ will be achieved via minimum decrease in energy at each iteration. Hence, the convergence time t will satisfy

$$\frac{E(v^o) - E(\overline{v})}{\lambda m} < t. \tag{7.20}$$

If we substitute the upper bound for $E(\overline{v})$ from (7.17), the inequality would still be valid

$$\begin{aligned} \frac{E(v^o) - E(\overline{\overline{v}}) - \lambda M}{\lambda m} &< t \\ \frac{E(v^o) - E(\overline{\overline{v}})}{\lambda m} - \frac{M}{m} &< t. \end{aligned} \tag{7.21}$$

Inspecting (7.21), we can see that the initial energy gap $E(v^o) - E(\overline{\overline{v}})$ affects the lower bound and it is important to start with a good initial motion-vector field. To make a fast start, we obtain an initial motion-vector field by exploiting non-spatial information as described in Chapter 6. If a truncated linear model is used for the penalty function, we will have $M = K$:

$$\frac{E(v^o) - E(\overline{\overline{v}})}{\lambda m} - \frac{K}{m} < t. \tag{7.22}$$

To reduce the lower bound on convergence time, one can either increase λ or K. But, λK determines the energy gap between the global and local minimum, which must be kept small. From (7.22), the first term is inversely proportional to λ, however the second term is linearly proportional to K. This shows that increasing λ will decrease the lower bound on t

[3] When we refer to the quality, either in terms of energy gap or convergence time, we refer to the worst-case quality.

faster than increasing K. Hence, in order to keep both the energy gap and the convergence time small, it would be wiser to choose a *larger* λ and a *smaller* K. Another factor that determines the lower bound is m, which is the minimum penalty between interacting blocks and it depends on the accuracy of the motion vectors: m is 1, 1/2, 1/4 for integer, half, and quarter-pixel accuracy, respectively. Since decreasing m will increase the lower bound, starting with a quarter-pixel accuracy will result in slow convergence. If the goal is to achieve quarter-pixel accurate motion vectors, starting with integer-pixel accuracy and then switching to half and quarter pixel accuracy sequentially during the iterations, will improve the convergence time compared to starting the iterations with quarter-pixel accuracy.

7.4 Primal-dual method for energy minimization

In the previous section, we presented a standard-move energy-minimization algorithm that operates by the minimization given in (7.12) for each block. To analyze its performance, we obtained an upper bound on its energy gap and a lower bound on its convergence time. Using these bounds, we discussed how to choose parameter values to speed up convergence to a local minimum that is close to the global minimum of a desirable energy function. In this section, we present a primal-dual method for motion estimation to further improve the performance.

As discussed earlier, an energy-minimization algorithm for motion estimation can fail because of a falsely designed energy function or convergence to a bad local minimum. The energy function involves two terms: a data term and a smoothness term. The data term relies on the brightness constancy assumption and combined with the translational-motion model becomes the gray-level matching error. The brightness constancy assumption is violated under noise and artifacts, and the translational motion model is not valid when there is rotation and zooming. Also, when there are repeating patterns or smooth regions in the scene, gray-level matching becomes ill-defined independent of the brightness constancy assumption and the motion model. To deal with the insufficiency of the data term

for gray-level matching, extra information is required. By assuming that the motion-vector field is smooth, extra information is obtained. The smoothness term tries to discriminate a random gray-level match from the true motion by penalizing variations in the motion-vector field. To achieve a smooth motion-vector field, the best gray-level match may not be chosen. Hence, the design of the smoothness term is crucial in the design of the energy function. By using a convex penalty function and correctly imposing the smoothness constraint, non-convexities in the data term can be remedied. Thus the smoothness term not only helps to find the true motion by the design of the energy function, but also helps with the convergence properties of the energy-minimization algorithm.

Prior works on energy minimization for stereo-disparity estimation algorithms fix the energy function at the beginning [66] [76] [67]. We present a joint approach in which the energy function is updated as the energy-minimization algorithm progresses by using the primal-dual method. The primal-dual method is commonly used in designing approximation algorithms since it yields combinatorial algorithms that have smaller energy gaps and convergence times [78]. The primal problem of energy minimization is transformed into its dual problem. After each iteration of the primal problem, the dual problem's approximate solution is computed. Then the smoothness term of the primal problem is updated using the relaxed-complementary-slackness conditions with the dual problem's solution. The dual problem tries to obtain the motion segmentation of the image, hence using the dual problem solution we can better impose the smoothness constraint by using not all neighboring blocks but only blocks that belong to the same motion object. With the help of the primal-dual method, we can use a larger λ since the smoothness constraint comes from a more reliable source. Larger λ improves the convergence time and the energy gap between the local minimum and the global minimum as discussed before.

In the next section we formulate a move as a linear program. The formulation allows for large/small or standard moves as dictated by the application.

7.4.1 Formulation of a move as a quadratic program

We define a move as a change of motion vectors to a particular candidate motion vector, ρ, similar to the α-expansion. A large move can change a group of block's motion vector to ρ, keeping all other vectors the same as the pre-move motion vector field v^*, while a standard-move can only change a single block's motion vector to ρ.

We formulate the move problem for candidate motion vector ρ as a standard-form quadratic problem (move-QP)

$$\min \sum_{b \in B} x_b D_b(\rho) + (1 - x_b) D_b(v_b^*) + \lambda \sum_{\{b,a\} \in \mathcal{N}} (1 - x_b)(1 - x_a) V_{b,a}(v_b^*, v_a^*)$$

subject to

$$\begin{aligned} x_b - x_a &\geq -p_{ba}, && \forall \{b, a\} \in \mathcal{N} \\ x_a - x_b &\geq -p_{ba}, && \forall \{b, a\} \in \mathcal{N} \\ x_b &\geq 0, && \forall b \in B \\ -x_b &\geq -1, && \forall b \in B, \end{aligned} \tag{7.23}$$

where v^* is the pre-move motion vector field, ρ is the candidate motion vector of the move, and p_{ba} is the probability of b and a to be on different objects/segments. x is the variable of the move-QP, which is in $[0 - 1]$. $x_b = 1$ indicates that $v_b = \rho$, and $x_b = 0$ indicates that $v_b = v_b^*$. For $x_b \in (0, 1)$, a fractional move is indicated, which can be interpreted as b takes on ρ with probability x_b and retains v_b^* with probability $(1 - x_b)$. The advantage of expressing the move problem as a linear program is that it enables fractional moves during the iterations.

The first term in (7.23) is the data term which is a weighted sum of gray-level matching errors with motion vectors ρ and v_b^*. The second term is the smoothness penalty for the blocks that retain their pre-move motion vectors. For any interacting block pair $\{b, a\}$ that retain its motion vectors (*i.e.*, $x_b = x_a = 0$), $\lambda V_{b,a}(v_b^*, v_a^*)$ is added as an additional cost to the energy. For a block that updates its vector to ρ, its smoothness penalty in the energy

becomes zero. Hence, two constraints are imposed to achieve spatial smoothness after the move: $x_b - x_a \geq -p_{ba}$ and $x_a - x_b \geq -p_{ba}$. They are standard-form linear constraints for $|x_b - x_a| \leq p_{ba}$. Neighboring blocks that are likely to move together are forced to move together depending on the probability, p_{ba}, of b and a to be on different objects/segments.

We next write the Lagrangian for the move-QP in (7.23),

$$\mathcal{L}(x,\beta,\gamma,\kappa) = \sum_{b\in B} x_b D_b(\rho) + (1-x_b)D_b(v_b^*) + \lambda \sum_{\{b,a\}\in\mathcal{N}} (1-x_b)(1-x_a)V_{b,a}(v_b^*, v_a^*)$$
$$+ \sum_{\{b,a\}\in\mathcal{N}} \beta_{ba}(-x_b + x_a - p_{ba}) + \beta_{ab}(-x_a + x_b - p_{ba}) + \sum_{\{b,a\}\in\mathcal{N}} -\gamma_b x_b + \kappa_b(x_b - 1), \quad (7.24)$$

where β, γ, and κ are positive *Lagrange multipliers* associated with their corresponding inequality constraints.

Lemma 7.4.1 *A binary solution*[4] *x of the move-QP corresponds to a move from v^* to v^{LP}.*

Proof By the definition of x, $v_b = \rho$ if $x_b = 1$, and $v_b = v_b^*$ if $x_b = 0$, which means a change of vectors to ρ while keeping all other vectors same as in v^*. Therefore, a binary x corresponds to a move from the previous motion vector field v^* to a new motion-vector field, v^{LP}, which is the solution of our quadratic program.

Using Lemma 7.4.1, and the Lagrangian in (7.24) yields:

Theorem 7.4.2 *The lagrangian of the move-QP with a binary solution x is equal to the energy, $E(v^{QP})$, if the lagrangian multipliers satisfy*

$$p_{ba} = 0 \qquad \forall\{b,a\} \in \mathcal{N},\ s.t.\ x_b = 1, x_a = 1 \qquad (7.25)$$

$$\beta_{ba} = 0,\ \beta_{ab} = 0 \qquad \forall\{b,a\} \in \mathcal{N},\ s.t.\ x_b = 0, x_a = 0 \qquad (7.26)$$

$$\beta_{ba} = 0,\ \beta_{ab}(1 - p_{ba}) = \lambda V_{b,a}(\rho, v_a^*) \qquad \forall\{b,a\} \in \mathcal{N},\ s.t.\ x_b = 1, x_a = 0 \qquad (7.27)$$

$$\beta_{ba}(1 - p_{ba}) = \lambda V_{b,a}(v_b^*, \rho),\ \beta_{ab} = 0 \qquad \forall\{b,a\} \in \mathcal{N},\ s.t.\ x_b = 0, x_a = 1 \qquad (7.28)$$

$$\gamma_b = 0, \kappa_b = 0 \qquad \forall b \in B, \qquad (7.29)$$

where v^{QP} is a move away from v^*.

[4] A binary solution is defined such that each component of x is either zero or one.

Proof From Lemma 7.4.1, a move from v^* to v^{QP} is given by a binary x. Let $\mathcal{S}^{0,0}$ be the set of block pairs $\{b,a\} \in \mathcal{N}$ and blocks $b, a \in B$, *s.t.* $x_b = 0, x_a = 0$. By using (7.26) and (7.29) in (7.24), the cost of $\mathcal{S}^{0,0}$ in the lagrangian denoted by $\mathcal{L}(x,\beta,\gamma,\kappa|S^{0,0})$ is given by

$$\mathcal{L}(x,\beta,\gamma,\kappa|S^{0,0}) = \sum_{b\in\mathcal{S}^{0,0}} D_b(v_b^*) + \lambda \sum_{\{b,a\}\in\mathcal{S}^{0,0}} V_{b,a}(v_b^*, v_a^*) \tag{7.30}$$

Since $x_b = 0$ denotes that the previous motion vector of b is preserved after the move, $v_b^{QP} = v_b^*$ for $b \in \mathcal{S}^{0,0}$. Substituting this in (7.30) to get

$$\mathcal{L}(x,\beta,\gamma,\kappa|S^{0,0}) = \sum_{b\in\mathcal{S}^{0,0}} D_b(v_b^{QP}) + \lambda \sum_{\{b,a\}\in\mathcal{S}^{0,0}} V_{b,a}(v_b^{QP}, v_a^{QP}) \tag{7.31}$$

Let $\mathcal{S}^{1,1}$ be the set of block pairs $\{b,a\} \in \mathcal{N}$ and blocks $b, a \in B$, *s.t.* $x_b = 1, x_a = 1$. By using (7.25) and (7.29) in (7.24), the cost of $\mathcal{S}^{1,1}$ in the lagrangian denoted by $\mathcal{L}(x,\beta,\gamma,\kappa|S^{1,1})$ is

$$\mathcal{L}(x,\beta,\gamma,\kappa|S^{1,1}) = \sum_{b\in\mathcal{S}^{1,1}} D_b(\rho) \tag{7.32}$$

Since $x_b = 1, x_a = 1$ indicates $v_b^{QP} = v_a^{QP} = \rho$, then $V_{b,a}(v_b^{QP}, v_a^{QP}) = V_{b,a}(\rho,\rho) = 0$ for any $b, a \in S^{1,1}$ by the definition of the smoothness penalty term, $V(.)$. Using these relations in (7.32)

$$\mathcal{L}(x,\beta,\gamma,\kappa|S^{1,1}) = \sum_{b\in\mathcal{S}^{1,1}} D_b(v_b^{QP}) + \lambda \sum_{\{b,a\}\in\mathcal{S}^{1,1}} V_{b,a}(v_b^{QP}, v_a^{QP}) \tag{7.33}$$

Let $\mathcal{S}^{1,0}$ be the set of block pairs $\{b,a\} \in \mathcal{N}$ and blocks $b, a \in B$, *s.t.* $x_b = 1, x_a = 0$. By using (7.27) and (7.29) in (7.24), the cost of $\mathcal{S}^{1,0}$ in the lagrangian denoted by $\mathcal{L}(x,\beta,\gamma,\kappa|S^{1,0})$ is

$$\mathcal{L}(x,\beta,\gamma,\kappa|S^{1,0}) = \sum_{b\in\mathcal{S}^{1,0}, x_b=1} D_b(\rho) + \sum_{a\in\mathcal{S}^{1,0},\ x_a=0} D_a(v_a^*) + \lambda \sum_{\{b,a\}\in\mathcal{S}^{1,0}, x_b=1, x_a=0} V_{b,a}(\rho, v_a^*) \tag{7.34}$$

$x_b = 1$ and $x_a = 0$ can be used to simplify the summation by substituting $v_b^{QP} = \rho$ and $v_a^{QP} = v_a^*$

$$\mathcal{L}(x,\beta,\gamma,\kappa|S^{1,0}) = \sum_{b\in\mathcal{S}^{1,0}} D_b(v_b^{QP}) + \lambda \sum_{\{b,a\}\in\mathcal{S}^{1,0}} V_{b,a}(v_b^{QP}, v_a^{QP}) \tag{7.35}$$

$\mathcal{L}(x,\beta,\gamma,\kappa|S^{0,1})$ can be derived similar to $\mathcal{L}(x,\beta,\gamma,\kappa|S^{1,0})$ by using (7.28).

Since $S^{0,0} \bigcup S^{1,1} \bigcup S^{1,0} \bigcup S^{0,1}$ includes all blocks in B and all interacting block pairs in $\mathcal{N}$:

$$\begin{aligned} \mathcal{L}(x,\beta,\gamma,\kappa) &= \mathcal{L}(x,\beta,\gamma,\kappa|S^{0,0}) + \mathcal{L}(x,\beta,\gamma,\kappa|S^{1,1}) + \mathcal{L}(x,\beta,\gamma,\kappa|S^{1,0}) + \mathcal{L}(x,\beta,\gamma,\kappa|S^{0,1}) \\ &= E(v^{QP}.) \end{aligned} \tag{7.36}$$

□

7.4.2 Sub-optimality of the energy

Theorem 7.4.2 shows that for a specific choice of the Lagrangian multipliers, the Lagrangian of the move-QP problem is equal to the energy under a binary move. The conditions for the Lagrangian multipliers for this equality may not be the best in terms of convergence and energy function design, because the Lagrangian multipliers are chosen without utilizing any information learnt from the iterations on the motion-vector field. By intelligently choosing the Lagrangian multipliers, contributions of the constraints in the Lagrangian can be better adjusted. This will especially improve the application of the smoothness constraints for blocks that are in the vicinity of a motion boundary.

Another justification for the sub-optimality of the energy from a theoretical point of view is as follows: As discussed before, an appropriate choice for hardware friendliness is a binary move, that is to evaluate the energy/Lagrangian[5] using a candidate motion vector ρ and update the motion vector to ρ if it results in a smaller Lagrangian. Hence, a binary solution of the move-QP problem is desired, but the optimal solution of the move-QP problem will be fractional as long as p_{ba}'s are not integers (*i.e.*, 0 or 1)[78]. An obvious technique to deal with this situation is to solve the linear program and then convert the solution to a binary solution via rounding, which is called *LP-rounding* [78]. Fractional moves are the solutions of the linear program and give tighter lower bounds than binary solutions. If the *rounding* operation does not increase the cost of the move-QP problem much, the approximate algorithm will perform well. Unfortunately, a linear problem is complex and difficult

[5] From this point on, energy and Lagrangian will be used exchangeably, but Lagrangian is used more often while discussing move-QP.

to implement in hardware. A more appropriate approach would be to use the dual of the move-QP problem in the design of the algorithm. In combinatorial optimization, this is called the *primal-dual schema* [**?**]. This technique constructs a feasible solution of the dual problem and an integer solution of the primal problem iteratively. Since a solution of the dual problem also provides a lower bound on the primal problem by the *weak duality*, one can compute a feasible solution for the dual problem first, which, then, can be used to find the corresponding primal solution via the *complementary slackness conditions* [**?**].

Our method differs from the primal-dual schema by the way the complementary slackness conditions (CSCs) are utilized. In the conventional primal-dual schema, CSCs are used to obtain the integer primal solution from the dual solution. However, we desire to design a *uniform* algorithm in the sense that we want to have one type of computation kernels, which iteratively update the motion vectors by substituting in the energy. Iterations of the same forms of energy function are more suitable for parallel implementations rather than an approach that involves minimizing an energy followed by solving the CSCs. In our method, CSCs are used to infer if a constraint of the move-QP is *tight* or *slack*. A slack constraint means it is satisfied and a tight constraint means it is satisfied with equality. Hence, by using the dual feasible solution and the CSCs, the Lagrangian multipliers can be computed. Compared to choosing the Lagrangian multipliers as specified by the conditions of Theorem 7.4.2 in an ad hoc manner, our primal-dual method uses the feasible dual solution to determine the Lagrangian multipliers, which is then used in the Lagrangian that is to be minimized. We present the dual problem of energy minimization in the following section.

7.4.3 The dual problem

We start with rewriting the move-QP problem in (7.23) in matrix form for a more compact representation. The primal problem is

$$\min_{x} c^T x + \bar{c}^T(\mathbf{1} - x) + (\mathbf{1} - x)^T W(\mathbf{1} - x) \tag{7.37}$$

subject to

$$Ax \geq -p$$

$$x \geq 0, -x \geq -1,$$

where c, $\bar{c}$, $x \in \mathbb{R}^{|B|}$, $W \in \mathbb{R}^{|B|\times|B|}$, $p \in \mathbb{R}^{|\mathcal{N}|}$, and $A \in \mathbb{R}^{|\mathcal{N}|\times|B|}$ such that $c_b = D_b(\rho)$ and $\bar{c}_b = D_b(v_b^*)$, $W_{ij} = V_{i,j}(v_i^*, v_j^*)$, $\forall\{i, j\} \in \mathcal{N}$, and $A_{ki} = 1$, $A_{kj} = -1$ for any k^{th} pair $\{i, j\} \in \mathcal{N}$. It is straightforward to construct the remaining vectors so that the above problem is equal to the original move-QP problem. This is a non-convex quadratic optimization problem since $W \not\geq 0$. The Lagrangian can be formed similar to (7.24):

$$\begin{aligned}
\mathcal{L}(x, \beta, \gamma, \kappa) &= c^T x + \bar{c}^T(\mathbf{1} - x) + (\mathbf{1} - x)^T W(\mathbf{1} - x) + \beta^T(-p - Ax) - \gamma^T x + \kappa^T(x - \mathbf{1}) \\
&= \left(c - \bar{c} - 2 \times \mathbf{1}^T W - A^T\beta - \gamma - \kappa\right)^T x + x^T W x - \beta^T p - \kappa^T \mathbf{1} \\
&= m^T x + x^T W x - \beta^T p - \kappa^T \mathbf{1}, \qquad (7.38)
\end{aligned}$$

where m is substituted for $c - \bar{c} - 2 \times \mathbf{1}^T W - A^T\beta - \gamma - \kappa$ for compactness. One can see that because of the way the constraints of the primal problem are utilized in the Lagrangian, the Lagrangian is always smaller than the primal problem's objective

$$\mathcal{L}(x, \beta, \gamma, \kappa) \leq c^T x + \bar{c}^T(\mathbf{1} - x) + (\mathbf{1} - x)^T W(\mathbf{1} - x), \qquad (7.39)$$

given that x is a feasible point in (7.37).

The dual function is a minimization of (7.38) on x,

$$\begin{aligned}
g(\beta, \gamma, \kappa) &= \inf_x \mathcal{L}(x, \beta, \gamma, \kappa) \\
&= \begin{cases} -1/4 m^T W^\dagger m - \beta^T p - \kappa^T \mathbf{1} & \text{if } W \geq 0, m \in \mathcal{R}(W) \\ -\infty & \text{otherwise,} \end{cases} \qquad (7.40)
\end{aligned}$$

which is derived by setting the gradient of the dual function to zero since it is a convex quadratic function of x. There are two conditions for the dual function to exist: W must be positive semi-definite and the vector m must be in the range $\mathcal{R}(W)$ of W. It is straightforward to show that the dual function provides lower bounds on the more complex primary problem's optimal value by observing (7.39) and (7.40).

Then the dual problem is the maximization of the dual function

$$\max_{\beta,\gamma,\kappa} -1/4 m^T W^\dagger m - \beta^T p - \kappa^T \mathbf{1} \tag{7.41}$$

subject to

$$W \geq 0,\ m \in \mathcal{R}(W).$$

Contrary to the primal problem, the dual problem is a convex optimization problem, however we are looking for a feasible solution of the dual problem, not necessarily the optimum solution. A feasible solution of the dual problem will be used to infer about the constraints of the primal problem.

The first feasibility condition requires that W is positive semi-definite. W is a symmetric matrix with $W_{ij} = V_{i,j}(v_i^*, v_j^*),\ \forall\{i, j\} \in \mathcal{N}$. Its elements are the penalty of variations in v for the interacting pairs in $\mathcal{N}$. Hence, positive semi-definiteness of W depends on the problem specific penalty values derived from the motion field v and the neighboring blocks that are assumed to interact. For example, it is easy to see that by using two immediate neighbors only, W will never be positive semi-definite regardless of what the penalty values are[6]. This shows that four-connectivity neighborhood is a minimum requirement for the dual problem to exist. An eight-connectivity neighborhood will make the dual problem more likely to exist, which will still depend on the v.

The second condition of dual feasibility is

$$m = (c - \bar{c} - 2 \times \mathbf{1}^T W - A^T \beta - \gamma - \kappa) \in \mathcal{R}(W)$$

Since $\mathbf{1}^T W$ is in the range of W if we set the remaining terms to zero, m will be in $\mathcal{R}(W)$. Hence,

$$\begin{aligned} 0 &= c - \bar{c} - A^T \hat{\beta} - \gamma - \kappa \\ \hat{\beta} &= A^\dagger (c - \bar{c} - \gamma), \end{aligned} \tag{7.42}$$

[6]It is straightforward to construct W using only two horizontal neighboring blocks and show that it is not positive semi-definite since W will be a symmetric tri-diagonal matrix with all its diagonal elements zero. Any x with alternating 1s and -1s as its elements will make $x^T W x \leq 0$

where $A^\dagger$ is the pseudo-inverse of A. κ is set to zero in the second equation, because the dual function that we want to maximize is a decreasing function of κ[7]. Since $A^\dagger(c - \overline{c})$ can be negative, the non-negativity assumption of $\hat{\beta}$ required for the derivation of the dual function can be violated. Hence, we need to clamp its components to zero from below. Since γ is unknown, one can choose γ to minimize the need for this clamping operation. At any rate, we can rewrite (7.42) to get the dual feasibiltiy condition (DFC)

$$\hat{\beta} = \left\lfloor A^\dagger(c - \overline{c}) \right\rfloor, \tag{7.43}$$

where $\lfloor\rfloor$ denotes the clamping operation applied to each component.

Using a feasible solution $\hat{\beta}$ of the dual problem, we can understand if the constraints of the dual problem are loose or tight, which can be used to re-design the energy function for the next iteration. This relation between $\hat{\beta}$ and the smoothness constraints, and also how it will be used for energy re-design is presented in the next section.

7.4.4 The primal-dual relation

First, we describe a method to solve for β. To have a unique solution for the DFC equation, $A^\dagger$ must be well-defined. However, by the way we defined the constraints in the move-QP each block pair $\{a, b\}$ has two constraints

$$x_b - x_a \geq -p_{ba}$$

$$x_a - x_b \geq -p_{ba}.$$

These two constraints constitutes two column vectors of A^T, which are identical (one points in the inverse direction). To solve for $A^\dagger = (AA^T)^{-1}A$, AA^T must be invertible. However, depending on the neighbor set $\mathcal{N}$, we may end up with a rank-deficient AA^T. To avoid this, it would be logical to eliminate one of the constraints, since only one of the constraints will be binding. For any of the two constraints to be binding, the left side of the inequality has to be negative. Hence, we can check if $x_b = 1, x_a = 0$ is more likely so that

[7] We note that $\kappa = 0$ does not necessarily produce the optimal solution of the dual problem.

$x_b - x_a \geq -p_{ba}$ can be eliminated or if $x_b = 0, x_a = 1$ is more likely so that $x_a - x_b \geq -p_{ba}$ can be eliminated.

We propose using the data terms to check the likelihood of $x_b = 1, x_a = 0$ and $x_b = 0, x_a = 1$, because at this stage we do not know what the solution of the problem x is . If $D_b(\rho) + D_a(v_a^*) > D_b(v_b^*) + D_a(\rho)$ then the first inequality will most likely be not binding and therefore can be discarded to achieve a unique β for the DFC[8].

To find a relation between β and move-QP constraints, we utilize the dual complementary slackness conditions[**?**]. If the complementary slackness conditions are satisfied, any feasible solution x of the primal and β of the dual problem are optimal. The dual complementary slackness condition is given below

Dual complementary slackness condition (DCSC)

$$\text{Either } \beta_{ba} = 0 \text{ or } A_{ba}x = -p_{ba}, \tag{7.44}$$

where $\{b, a\}$ is any block pair and A_{ba} is the row for the constraint associated with β_{ba}. The complementary slackness conditions are important in the design of efficient approximation algorithms for complex problems, which try to improve x and β by modifying them in a way that more of their components satisfy the complementary slackness conditions[**?**].

Our goal for applying the primal-dual method is to re-design the energy function, which corresponds to the lagrangian of the primal problem. The lagrangian is showed to be equivalent to the energy function under the conditions on the lagrangian multipliers given in Theorem 7.4.2. However, our choice of the constraints were imposed for all the blocks pairing with a block, independent of the motion segmentation since the unknown motion-vector field v is what we want to estimate. Fortunately, $\hat{\beta}$ reveals some information on the *connectedness* of blocks with their neighbors based on their motion by the use of DCSC. DCSC dictates that if $\hat{\beta}$ is non-zero, then the constraint is binding and must be satisfied

[8]For a standard-move algorithm elimination of one of the two equations between two blocks happens inherently, which will be discussed in Section 7.5.1

with equality. If the constraint was relaxed, the interacting blocks will choose different directions. This implies that the two blocks are on different objects or object segments that are moving in different directions. Hence, the constraint that is binding is conflicting with our smoothness of v assumption, and should be imposed less by decreasing its contribution in the energy. To this end, the lagrangian multipliers must be inversely related to $\hat{\beta}$, for example by multiplying with a function $f(\hat{\beta})$ that is decreasing with $\hat{\beta}$ and has range $[0, 1]$. Hence, by DCSC we propose to update the β values as given below

$$\beta_{ba}^{DCSC} = f(\hat{\beta}_{ba})\beta_{ba}, \tag{7.45}$$

where $\hat{\beta}$ is a feasible solution of the dual problem given in (7.43), and f is a decreasing function of $\hat{\beta}$. Replacing lagrangian multiplier β with β^{DCSC} and substituting $\kappa = 0$ from the solution of the dual problem, lagrangian in (7.24) becomes

$$\begin{aligned}\mathcal{L}(x, \beta, \gamma, \kappa) = &\sum_{b \in B} x_b D_b(\rho) + (1 - x_b) D_b(v_b^*) + \lambda \sum_{\{b,a\} \in \mathcal{N}} (1 - x_b)(1 - x_a) V_{b,a}(v_b^*, v_a^*) \\ &+ \sum_{\{b,a\} \in \mathcal{N}} \beta_{ba}^{DCSC}(-x_b + x_a - p_{ba}) + \beta_{ab}^{DCSC}(-x_a + x_b - p_{ba}) + \sum_{\{b,a\} \in \mathcal{N}} -\gamma_b x_b, \end{aligned} \tag{7.46}$$

If β values are chosen to satisfy the conditions of Theorem 7.4.2, the lagrangian will boil down to the energy with the addition of $f(\hat{\beta}_{ba})$ multiplying the smoothness constraint betwen b and a,

$$E(v) = \sum_{b \in B} D_b(v_b) + \lambda \sum_{\{b,a\} \in \mathcal{N}} f(\hat{\beta}_{ba}) V_{b,a}(v_b, v_a). \tag{7.47}$$

The above energy formulation does not isotropically enforce a smoothness constraint, but adapts the weights of constraints in the energy with information derived from the data via the dual problem. This will enable us to obtain an energy function that is more powerful to explain desired motion vectors across object and segment boundaries.

7.5 A hardware-friendly standard-move algorithm

As mentioned in Section 7.3, a standard-move can change one block's motion vector. standard-move algorithms have less computational complexity and are easier to implement,

rendering them as perfect candidates for use in consumer products. In this section, we will design a hardware-friendly standard-move algorithm for motion estimation based on the results derived in previous sections.

To achieve a real-time algorithm, we want to minimize complexity whenever possible. A four-connectivity neighborhood (*i.e.* horizontal and vertical neighbors) is the smallest neighborhood for the dual problem to exist as obtained in Section 7.4.3. Therefore, we choose to use horizontal and vertical neighbors for the smoothness cost, which is denoted by $\mathcal{N}_4$.

In the following we discuss the necessary simplifications to achieve a standard-move version of the lagrangian given in (7.46) that uses $\mathcal{N}_4$.

7.5.1 Simplifications on the energy

Similar to Section 7.3, we start with removing the terms in the lagrangian that are not affected by a standard-move. We only keep terms in (7.47) that involve b to find the energy contributed by it:

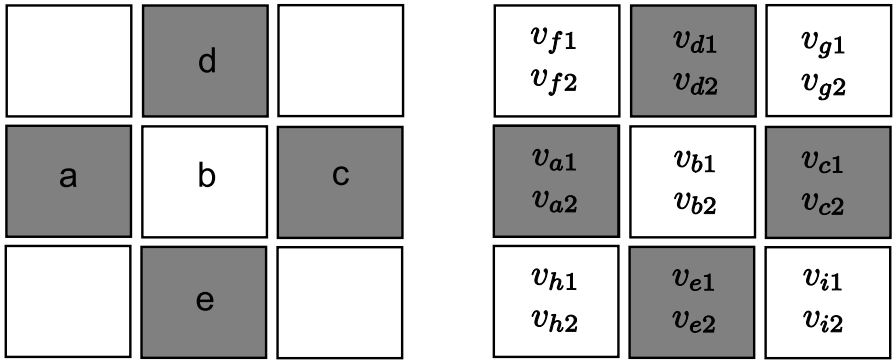
$$E_b(v) = D(v_b) + \lambda \sum_{i \in \mathcal{N}_4} f(\hat{\beta}_{bi}) V(v_b, v_i), \tag{7.48}$$

where we discontinue to use the subscript of $D()$ by assuming that the data term for each block is the same (*e.g.* sum of absolute deviations). Also, the subscript of $V()$ is dropped since its response to a specific block pair $\{b, a\}$ is captured in $f(\hat{\beta}_{ba})$, which is satisfactory for our purposes.

To obtain the best standard-move for b, all possible moves should be searched to find the one that minimizes $E_b(v)$, hence, optimal standard-move for b is given by

$$\arg\min_{\rho \in S} D(\rho) + \lambda \sum_{i \in \mathcal{N}_4} f(\hat{\beta}_{bi}) V(\rho, v_i). \tag{7.49}$$

Using $\mathcal{N}_4$ as given in Figure 7.1(a), there can be eight constraints obtained from the four pairs (*i.e.* two per pair). Since we are designing a standard-move algorithm, for each $b \in B$ and $\rho \in S$, we should see if x_b is one or zero ($x_b = 1$ denotes that b will change its vector from v_b^* to ρ). By allowing for the four neighbors to have fractional x values, while



(a) Neighbors of b for smoothness (b) 18 candidate vectors

Figure 7.1. Blocks a, c, d, e are used to enforce smoothness, while best two vectors of blocks in $\mathcal{N}_8$ in addition to best two vectors of b form S', which is the set of candidate vectors for the standard-move of b. The blocks used for smoothness is shaded gray.

substituting $x_b = 1$ in the lagrangian implies that only one of the two constraints can be binding, which are

$$x_a - x_b \geq -p_{ba}$$

$$x_c - x_b \geq -p_{bc}$$

$$x_d - x_b \geq -p_{bd}$$

$$x_e - x_b \geq -p_{be}.$$

The constraint matrix A, and $A^\dagger$ are

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}, \qquad A^\dagger = \begin{bmatrix} 0.8 & -0.2 & -0.2 & -0.2 & -0.2 \\ -0.2 & 0.8 & -0.2 & -0.2 & -0.2 \\ -0.2 & -0.2 & 0.8 & -0.2 & -0.2 \\ -0.2 & -0.2 & -0.2 & 0.8 & -0.2 \end{bmatrix}, \tag{7.50}$$

where A's columns from left to right represent x values for $a, c, d, e,$ and b, respectively. Substituting (7.50) in (7.43)

$$\begin{bmatrix} \hat{\beta}_{ba} \\ \hat{\beta}_{bc} \\ \hat{\beta}_{bd} \\ \hat{\beta}_{be} \end{bmatrix} = \frac{1}{5} \begin{bmatrix} \lfloor 4d_a - d_b - d_c - d_d - d_e \rfloor \\ \lfloor 4d_c - d_a - d_b - d_d - d_e \rfloor \\ \lfloor 4d_d - d_a - d_b - d_c - d_e \rfloor \\ \lfloor 4d_e - d_a - d_b - d_c - d_d \rfloor \end{bmatrix}, \tag{7.51}$$

where d_a is $c_a - \bar{c}_a$, and d_b is $c_b - \bar{c}_b$, etc.

The analysis of $\hat{\beta}$'s response to different motion-vector scenarios shows that it changes consistently with our expectations. For example, if the candidate vector ρ does not apply well to block a, c_a will probably be larger than $\bar{c}_a$ (*i.e.* $d_a > 0$), and if at the same time it applies well to the other blocks (b, c, d, e) (*i.e.* $d_b, d_c, d_d, d_e < 0$), than $\hat{\beta}_{ba}$ is significantly larger than zero. By the DCSC, this implies that its corresponding constraint is binding. Following the arguments in Section 7.4.4, this is reflected in the design of the energy function by decreasing its weight via (7.45). This means that the vectors of c, d, and e are mostly utilized to impose the smoothness constraint on v, not a. As a second example, assume that ρ does not apply well to b, c, d, and e, which would require d_b, d_c, d_d, and d_e to be greater than zero, which, in turn, would indicate a smaller $\hat{\beta}_{ba}$. This will imply that the corresponding constraint between b and a will less likely be binding, hence its weight will be larger than in the previous example. To summarize, in the first example, a was on a different object than b and it moves with a vector other than ρ, but c, d, and e were on the same object with b and they move with a vector that is close to ρ. Hence, they were used to derive the required smoothness information, not a. In the second example, none of the five blocks' motion could be explained by ρ, and therefore the smoothness constraints were applied isotropically. One final note is that if a candidate vector ρ applies well to b ($d_b < 0$), the chances of decreasing the weight of the constraints goes up (*i.e.* a good candidate will less likely be rejected because of an overweighted smoothness cost).

7.5.2 Further simplifications on the algorithm

In the previous subsection, we have derived the energy function for a standard-move and its weights as given in (7.49) and (7.51). The smoothness cost is adapted according to the information obtained from the dual problem that runs on $\mathcal{N}_4$ to account for cases where smoothness can not be enforced isotropically. This requires that only blocks in $\mathcal{N}_4$ are used to enforce smoothness. Although these simplifications reduce the computations, we need further simplifications for speeding up a hardware implementation. Next, we present some simplifications on the algorithm.

1. standard-moves are the least complex moves, but we still need to reduce computations since standard-move algorithms are $O(|S||B|)$ complexity as discussed in Section 7.2, and $|S|$ is a large number because the search window S for the vectors is desired to be large to cover big motions. The number of blocks, $|B|$, is determined by the image size, however, the number of possible motion-vectors can be reduced intelligently. In Section 7.1.5, we discussed starting with an initial motion-vector field, v^o, that uses non-spatial prior information from previous frame or a coarser resolution. Starting the standard-moves with a good v^o will speed-up convergence to a good local minimum while avoiding getting stuck at bad local minimums. Using the hypothesis testing techniques presented in Chapter 6, one can search for an initial motion-vector field by minimizing (6.20). This is a bi-criterion cost function that consist of a data term for block-matching and a term for measuring the distance from a previously obtained bias vector, which adaptively penalizes deviations from the bias vector. Fortunately, one can use the results of this vector search in S for creating a reduced candidate vector set S' that is significantly smaller than S. Since, an exhaustive search in a window for creating the initial motion-vector field samples the cost curve, it reveals information about the location of good candidate vectors. By keeping track of vectors that produce a small cost in the search for v^o, S can be limited to S'. To achieve a small $|S'|$, a simple yet effective method is to represent

the cost curve of (6.20) by picking two best vectors (v_{b1}, v_{b2}) from this minimization for b. Oftentimes, since the blocks are smaller than objects, the same vector can describe the motion for more than one block in a locality. Hence, we populate S' by also using best two vectors from blocks in its eight-connectivity neighborhood, which makes $|S'| = 8 * 2 + 2 = 18$ in total, as shown in Figure 7.1(b). This introduces a scalability to our solution since S' is fixed in size, independent of the search window size $|S|$. The standard-move problem given in (7.49) for b becomes

$$\arg\min_{\rho \in S'} D(\rho) + \lambda \sum_{i \in \mathcal{N}_4} f(\hat{\beta}_{bi}) V(\rho, v_i). \tag{7.52}$$

After the first round of standard-moves over all blocks, in the second round S' is updated using the results of the previous round instead of using previous S' one more time. As standard-moves refine v, candidate vectors should also be improved by using the updated vectors from the previous round so that moves to better candidates are considered.

2. Avoiding data latency and/or synchronization is crucial for hardware-friendliness. Neighboring blocks' vectors are used in the smoothness cost or they are used as candidates to search for the best standard-move. This can be achieved with a recursive implementation so that a block waits on some of its neighboring blocks to update their vectors. However, this will introduce delays in the computations. For example, for an implementation on a GPGPU, one needs to minimize synchronizations between different thread blocks, which requires access to device memory (DRAM), and maximize use of on-chip memory (cache). To do so, it is better to launch a kernel for each round of standard-moves. At each round, thread blocks will bring the data from DRAM to cache, and work on it independent of the other thread blocks and other threads in the same thread block. This will minimize the synchronization needs and idle waiting times. The results will be written back to DRAM afterwards. By launching a new kernel for the next round, all thread blocks will be using the updated

v without having to synchronize between thread blocks, which is a difficult and an inefficient process since GPUs are designed to exploit parallel algorithms that does not need much flow control. Therefore, we make another simplification by using v', which is the previous round's motion-vector field, for finding the moves to create v'', which is the next round's motion-vector field. This way, the algorithm will be iteratively working on v. Hence, each round of standard-moves over all the blocks is called an *iteration*.

3. Another simplification would be to reduce the search window, S, used for creating the initial motion-vector field, v^o, to further reduce computations. However, reducing the search window used in v^o creation will restrict our candidate set, S', for standard-moves to smaller motion vectors since S' is derived from the results of S. To be able to find large motion vectors while reducing S, a hierarchical motion search can be utilized. Starting with a coarser resolution and a smaller search window, standard-move iterations can be formed to refine vectors at the coarser resolution. The coarse resolution vectors can then be used to center search windows in the finer resolution. An important problem that would arise from using a hierarchical motion search is to choose the course resolution vector that is going to be used to center the fine-resolution search. For example, using down-sampling by two to create a half-resolution and a full-resolution image, each full-resolution block will occupy one quarter of a half-resolution block, which can cover different motion segments and objects in it. To solve this problem, half-resolution blocks must be partitioned into quarter-blocks and their vectors can be refined using standard-move iterations to refine the motion-vector field. After these refinements, quarter block vectors from the half-resolution can be used to center the search windows of full-resolution blocks. Since a quarter block in the half-resolution will have the same size with the corresponding block in the full-resolution, the centering approach will be more robust.

7.5.3 Some heuristics for fast convergence

In the previous sections, we managed to obtain a low-complexity iterative energy-minimization algorithm, which also updates the energy via the dual problem as iterations progress. In this section we will introduce some heuristics for speeding up the convergence time to a *good* local minimum.

1. To restrict S to S' for finding the best standard-move, we proposed to use best two vectors of blocks in $\mathcal{N}_8$, instead of trying all possible vectors in the search window S. To make this reduction to two best vectors more robust, one can enforce a minimum distance between them. This way, we will better subsample the cost in the search window and avoid picking two *bad* local minimums that are close to each other.

2. S' consists of 18 candidate vectors, and 9 of these vectors are second best vectors that minimize the bi-criterion cost to utilize non-spatial information from the previous frame or resolution. Candidate vectors are used to find the best standard-move by minimizing the energy in (7.52). The smoothness cost component of the energy is obtained using the vectors of blocks in $\mathcal{N}_4$. But using the first-best vector, v_{a1}, of a block $a \in \mathcal{N}_4$ in the smoothness cost, we will be favoring v_{a1} against its second-best vector, v_{a2}, since $V(v_{a1}, v_{a1}) = 0$, but $V(v_{a1}, v_{a2}) \geq 0$. Due to violations of our translational-motion model and brightness constancy assumption, it is sometimes the case that second-best vector is actually the true vector. But as v is refined it may turn out to be the first-best vector after the iterations are finished. Hence, one needs to enable the second-best vector to survive the picking-best-two-vector process throughout standard-move iterations by permitting them to contribute in the smoothness cost too. By doing so, a local minimum caused by more than one wrong vector in a locality can be overcome as iterations progress, and this enables a faster convergence to a better local minimum. However, the weight of the second-best vectors must be less than the first-best vectors, since first-best vectors are more often the true motion

vectors. To do so, the standard-move problem in (7.52) for b is changed to

$$\arg\min_{\rho \in S'} D(\rho) + \lambda_1 \sum_{i \in \mathcal{N}_4} f(\hat{\beta}_{bi}) V(\rho, v_{i1}) + \lambda_2 \sum_{i \in \mathcal{N}_4} f(\hat{\beta}_{bi}) V(\rho, v_{i2}), \tag{7.53}$$

where $\lambda_1 > \lambda_2$.

3. As discussed in Section 7.3, choice of λ and K is important to minimize the upper-bound of the energy gap between the local minimum, $\bar{v}$, reached at the end of iterations, and the global minimum, $\bar{\bar{v}}$. Also, the lower-bound of the convergence time from v^o to $\bar{v}$ depends on λ and K. The derivations showed that a larger λ and a smaller K should be desired in the choice of the parameters to improve these two important performance measures. However, starting with a large λ may force moves to a bad local minimum since v^o may have bad local minimums and may cause the rejection of all moves to a better local minimum. For example, when most of the vectors in a locality are wrong, smoothness cost will be punishing true motion vectors by dominating the data term due to a large λ. One can avoid this situation by starting with a smaller λ that allows moves to true motion vectors, and increase it as the iterations progress to attain a smaller upper-bound on the energy gap, and a smaller lower-bound on the convergence time. One should note that increasing λ with the iteration count can also be justified by using the equivalency between the energy and a posterior probability as described in Section 7.1.3. As the iterations progress and v is refined, the prior information becomes more reliable, and the precision (inverse of scale/variance) of the prior distribution can be increased by using a larger λ. However, λ can not be increased indefinitely, since this will cause the smoothness cost to dominate the data term in the energy. Hence, one can choose λ as a function of iteration number, t, as below

$$\lambda_1(t) = \lambda_1 \lceil t \rceil^T, \quad \lambda_2(t) = \lambda_2 \lceil t \rceil^T, \tag{7.54}$$

where $\lceil \rceil^T$ represents clamping from above to T. By increasing the weight of the

smoothness cost, which is a convex term, the likelihood of getting stuck at a local minimum is reduced.

4. As discussed in the third simplification of the previous subsection, hierarchical motion search requires the coarse-resolution blocks to be partitioned into quarter-blocks. By refining the quarter-blocks using standard-move iterations before passing to the finer resolution, motion search in the fine resolution becomes more reliable. To improve the quality of quarter-block vectors, one can use candidate vectors from the previous frame's motion vectors. Using only vectors of the coarse resolution blocks to produce quarter-block vectors, it would be difficult to find the motion of objects that are smaller in size. In such a case, a block will be covering more than one object, hence its vector can be corrupted. But a previous frame's final motion vectors could have already solved this problem by partitioning into smaller blocks. Also, using candidate vectors from the previous frame will increase the robustness of the algorithm since the motion of objects tend to change smoothly in time.

The first and the fourth heuristics help to speed up the convergence by increasing the quality of the candidates for standard-moves, the second and third heuristics modify the energy function to to be able to choose better moves, and also reduce the weight of the non-convexities by increasing the weight of the convex terms.

7.5.4 Proposed motion-estimation algorithm

In this section, we present the complete algorithm by bringing together the necessary equations, simplifications, and heuristics from the previous sections. A block diagram is given in Figure 7.2, and the pseudo-code is given in Algorithm 2. Our proposed motion-estimation algorithm is a hierarchical motion-estimation algorithm that uses a full-resolution and a half-resolution image produced by down-sampling the full-resolution image by two. The algorithm uses motion information from the previous frame to be used in half-resolution's

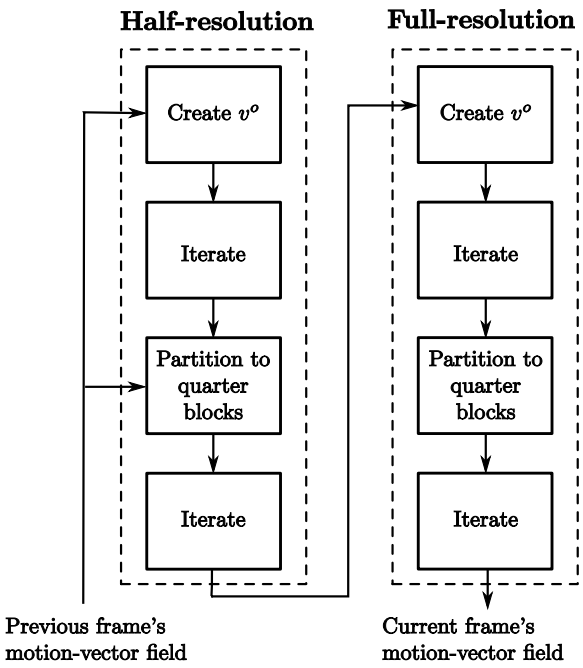


Figure 7.2. Block diagram of the algorithm

motion search, and candidate set creation for partitioning to quarter-blocks. After the half-resolution vectors are finalized, full-resolution motion-vector processing starts. Below, we give some explanations with the corresponding line numbers in the pseudo-code given in Algorithm 2.

Line 5: For each half-resolution block, a motion search is performed to pick the two best motion vectors with a minimum distance criteria to minimize a cost. Bias vectors in the cost are chosen from the previous frame using hypothesis testing, and their weights in the cost are adapted with how well they apply to a block. Use of bias vectors help in the cases of low detail areas and repeating structures, where brightness constancy assumption does not work well, and also under rotation and zooming, where translational-motion model is not valid. These issues are discussed in Chapter6.

Lines 7-12: The initial motion-vector field v^o is refined using N standard-move iterations. Candidate vectors in S' is used to pick the best-two standard-move for each b. S' consists of 18 vectors obtained from a block's and its eight-connectivity neighbors' best-two vectors. Smoothness cost is computed using the previous iteration's vectors to speed up

execution time on hardwares, which is discussed in Section 7.5.2. Smoothness cost is derived from blocks in four-connectivity neighborhood and weights of the neighboring blocks is adapted using information from the dual problem. Second-best vectors also contribute to the smoothness cost, but to a lesser extent than first-best vectors ($\lambda_1 > \lambda_2$). These choices are made to enable a faster convergence, and breaking away from bad local minimums, which are discussed in Section 7.5.3.

Algorithm 2: An iterative motion-estimation algorithm

Input: Previous and current images: p, c
Output: Final motion-vector field: $\overline{v}$

1 Initialize λ_1, λ_2, K, T, d^{min}, N;
2 Set $R = \{Half\text{-}resolution, Full\text{-}resolution\}$;
3 **foreach** $r \in R$ **do**
4 Initialize $D()$ using p and c for r;
5 Pick v^o_{b1}, v^o_{b2} s.t. $|v^o_{b1} - v^o_{b2}| > d^{min}$ by minimizing (6.20) for each block b;
6 Set v' to v^o;
7 **while** $t < N$ **do**
8 **foreach** $b \in B$ **do**
9 $\{v''_{b1}, v''_{b2}\} =$
 $\arg\min_{\rho \in S'} 2 \quad D(\rho) + \lambda_1 \lceil t \rceil^T \sum_{i \in \mathcal{N}_4} f(\hat{\beta}_{bi}) V(\rho, v'_{i1}) + \lambda_2 \lceil t \rceil^T \sum_{i \in \mathcal{N}_4} f(\hat{\beta}_{bi}) V(\rho, v'_{i2});$
10 **end**
11 Set v' to v''
12 **end**
13 Update v' by partitioning into quarter-blocks;
14 **while** $t < N$ **do**
15 **foreach** $b \in B$ **do**
16 $\{v''_{b1}, v''_{b2}\} =$
 $\arg\min_{\rho \in S'} 2 \quad D(\rho) + \lambda_1 \lceil t \rceil^T \sum_{i \in \mathcal{N}_4} f(\hat{\beta}_{bi}) V(\rho, v'_{i1}) + \lambda_2 \lceil t \rceil^T \sum_{i \in \mathcal{N}_4} f(\hat{\beta}_{bi}) V(\rho, v'_{i2});$
17 **end**
18 Set v' to v''
19 **end**
20 **end**
21 Set v'

Line 13-19: Each block is partitioned to quarter-blocks so that a half-resolution quarter-block matches with a full-resolution block in size. This will increase the reliability of the centered motion-search in full-resolution. Also, partitioning to quarter-blocks increases

the quality of half-resolution vectors in general, since with a smaller block size block-based translational-motion model is less problematic for rotation, zooming, and motion boundaries. Vectors from the previous frame are also used as candidates during the partitioning, which increases the robustness of the algorithm in general and especially helps in the case of small objects/segments. These issues are discussed in Section 7.5.3. After the partitioning, standard-move iterations are again applied to refine further, before passing to full-resolution.

7.6 Experiment Results

In this section, we present experiment results using the proposed algorithm to investigate good parameter values for the energy, and its minimization via standard-move iterations. We do not use a quantitative measure for assessing the quality of the produced motion-vector field, but rather give visual results depicting the motion-vector field, and try to analyze its sensitivity and behavior with respect to different algorithmic blocks and their corresponding parameters. We believe the best quality assessment is human-eye inspection, since any quantitative quality measure tries to understand if the motion-vector field is good or not and this is in fact another formulation of the motion-estimation problem we are trying to solve[9].

The implementation used to produce the results operate on images down-sampled by two via pixel dropping. The down-sampled images are full-resolution images, another down-sampling is performed after low-pass filtering, which produces the half-resolution images. Hence, an 8x8 block in half-resolution corresponds to a 32x32 block in the original scale: 64 pixels are used to compute the SAD measure instead of 1026 pixels.

[9] A variant of this algorithm will be used in NVIDIA's drivers for frame-rate up-conversion, and the results presented in this section are chosen based on large number of experiments.

7.6.1 A walk-through of the algorithm

We start with a walk-through of the algorithm by presenting intermediate motion-vector field results. The motion-vector field is used to modulate the chroma signal in the image, which makes it easier for the observer to asses the quality.

The half-resolution motion-vector field for the full-blocks (HR-FULL) is given in Figure 7.3(a), which is refined using standard-move iterations on top of the initial motion-vector field created using non-spatial information. At this stage, motion vectors are coarse, and the object boundaries are completely smeared. However, the motion-vectors are robust in the sense that they show the general trend of an object's motion without any outliers.

Figure 7.3(b) shows half-resolution quarter-blocks' (HR-QUARTER) motion-vectors, which are produced by partitioning full-blocks into quarter-blocks using HR-FULL motion-vectors and previous frame's motion-vectors. Quality is improved and motion boundaries become more visible. One can see that some "colors" that are not visible in HR-FULL appears in HR-QUARTER. The reason is that candidate sets for partitioning also include previous frame's motion-vectors. These vectors may not be available in HR-FULL, since they probably belong to small object segments that can not survive in HR-FULL, however, they were available in the previous frames's final motion-vector field.

Using HR-QUARTER, the motion-vector field of the full-blocks in full-resolution (FR-FULL) is created. First, a motion-search that is centered around and biased towards HR-QUARTER motion-vectors is performed. The initial motion-vector field created as a result is then refined using standard-move iterations, which can be seen in Figure 7.3(c). One can see that the "color" is richer than HR-QUARTER because motion-search at full-resolution stage is performed on blocks that are four times smaller than the blocks in half-resolution. However, with smaller blocks SAD measure sometimes becomes less reliable since a smaller block random matching are more likely to occur. For example, a purple block appear in the occlusion area between the two left-most soldiers.

Next, full-resolution quarter-blocks' (FR-QUARTER) motion-vector field given in Figure 7.3(d) is obtained using FR-FULL motion-vectors via partitioning followed by refinement iterations. The motion boundaries is improved significantly compared to FR-FULL boundaries. Fine objects like the middle soldier's rifle has been resolved because the rifle is too thin for blocks in FR-FULL and therefore its motion was mixed with the background motion in FR-FULL motion-vector field. However, FR-QUARTER block sizes were small enough to distinguish the rifle from the background, hence the true motion-vectors from the body of the soldiers is *diffused* onto the rifle via standard-move refinement iterations. Also, the purple block at the intersection of two left-most soldiers is corrected due to a better applicability of block-based translational-motion model on smaller blocks.

The final step is one more partitioning and refinement to get quarter-quarter blocks in full-resolution (FR-QUARTER^2). As in the previous stage, motion boundaries are improved even further with finer block sizes. The motion boundaries follow the object boundaries closely and one can guess the objects by inspecting the motion-vector field in FR-QUARTER^2 given in Figure 7.3(e). In Figure 7.3(f), a composite image is obtained by combining the motion modulated chroma channels with the luminance component of the original image.

7.6.2 Sensitivity analysis for λ

The choice of smoothness cost coefficient, λ, is crucially important since it determines the contribution of our prior information at the expense of data. Previously, by analyzing our standard moves we deduced that large λ values are necessary to minimize the inevitable energy gap between the converged local minimum and the global minimum and also to minimize the convergence time. However, large λ values may prevent the minimization to break away from bad local minimums of the energy in the initial motion-vector field. An example for this situation is given in Figure 7.5 using HR-QUARTER stage results. We use the condition $\lambda_1 = 4\lambda_2$, since $\lambda_1 > \lambda_2$ is required as discussed before. For $\lambda_2 = 12|BS|$ ($|BS|$ is the block size), some strong bad local minimums created in the initial motion-vector field

prevent the algorithm to jump to other solutions. These bad local minimum are created due to the ill-posed nature of intensity matching. For example bad local minimums can be created due to repeating blinds, which may favor vectors that match a blind with another blind, as seen in the case of motion-vectors represented by green areas in Figure 7.4(a) and Figure 7.4(b). Also, bad local minimums can be created due to other sources of deficiency in block matching. For example, motion-vectors tend to slip along a strong intensity edge caused by a small eigen-value of the correlation matrix of the intensity gradients in a locality (*i.e.* the aperture problem). In Figure 7.4(a), Figure 7.4(b), and Figure 7.4(c) cyan regions on the blind appear because motion-vectors slip vertically parallel to the blinds. Both of these two local minimums become weak if λ is chosen small. For $\lambda = 0.5$ all the standard-moves succeed in breaking away and converge to a better local minimum shown in Figure 7.4(d).

7.6.3 Sensitivity analysis for K

Our proposed algorithm thresholds the motion-vector differences to K for smoothness cost computation. Small K values improve the energy gap and convergence time in addition to preserving motion boundaries as discussed before. However, too small K values does not distinguish between small and big motion-vector differences and therefore is only good when the solution of the minimization problem is restricted to binary values. In Figure 7.5(a), results are produced by setting K to 64, which causes the motion of the background seen through the jeep's glasses to be merged with the jeep's motion. However, its motion-vectors are preserved by using $K = 4$, as seen in Figure 7.5(b). By setting $K = 1$ the motion boundaries are preserved slightly better at the cost of preserving wrong motion boundaries: the motion-vector field looks less smooth, and some wrong motion-vectors survive.

7.6.4 Sensitivity analysis w.r.t the number of iterations

In our implementation we kept N, the number of iterations constant for all the algorithmic blocks, namely, HR-FULL, HR-QUARTER, FR-FULL, FR-QUARTER, and FR-QUARTER^2. For lowering the computational complexity, it is better to choose larger N when there are fewer pixel blocks, and to choose a smaller N, when there are more pixel blocks. If we denote the number of blocks with n_1, n_2, n_3, n_4, n_5 with respect to the order given above, then $n_5 = 4n_4$, $n_4 = 4n_3$, $n_3 = n_2$, $n_2 = 4n_1$, hence, the cost of 1 iteration in FR-QUARTER^2 is about the cost of 64 iterations in HR-FULL. By choosing more iterations in lower stages of the algorithmic flow, we can get computational complexity reductions. In addition to this, the overall quality of the motion-vector field may also be improved since each algorithmic block uses the results of the previous algorithmic block. Therefore, a quality improvement in the earlier stages of the algorithm can propagate to later stages and improve the final quality significantly. Figure 7.6(a) is produced by fixing the number of iterations of all stages to 5 ($n_5 = n_4 = n_3 = n_2 = n_1 = 5$). Using the above relation on the number iterations at different stages, it can be seen that one iteration of FR-QUARTER is equivalent to three iterations of HR-QUARTER plus four iterations of HR-FULL. Hence, we can decrease n_2 to 4, and increase n_4 to 8 and n_5 to 9 and still have approximately the same computational complexity. The results for this configuration is given in Figure 7.6(b). The quality has been improved, and some bad local minimum created at occlusion areas has been removed via more iterations. Since our proposed energy function changes with the help of $f(\hat{\beta}_{bi})$ and $\lceil t \rceil$ terms as the iterations progress, more iterations succeed to modify the energy to eliminate the observed bad local minimums in Figure 7.6(a).

7.6.5 Analysis of $f(\hat{\beta}_{bi})$

The spatial smoothness cost contributed from each block i in the neighborhood of b is modified with $f(\hat{\beta}_{bi})$ so that blocks on different objects and segments will contribute less to the energy. In addition to improving the quality around motion boundaries this will also speed up the convergence by formulating an energy function that can better fit to data by

modeling motion boundaries. The results in Figure 7.7(a) and Figure 7.7(c) are produced by setting $f(\hat{\beta}_{bi}) = 1$, while Figure 7.7(b) and Figure 7.7(d) are produced by computing $\hat{\beta}_{bi}$ from (7.51) as a feasible solution of the dual problem and choosing $f(x) = \frac{C-[x]_0^C}{2C} + 0.5$, where C is a constant and $[]_0^C$ denotes clamping from below to zero and from above to C. $f(x)$ is designed to have a minimum at 0.5 so that all neighbors contribute even when they have high $\hat{\beta}$ values, which takes into account the fact that the dual may not exist and therefore $\hat{\beta}$ values are not reliable. As can be seen from the figures, quality of the results has been improved with the use of $f(\hat{\beta}_{bi})$ including the background seen through the jeep's windows or right above the jeep, and the region between the wheels.

7.7 Conclusions

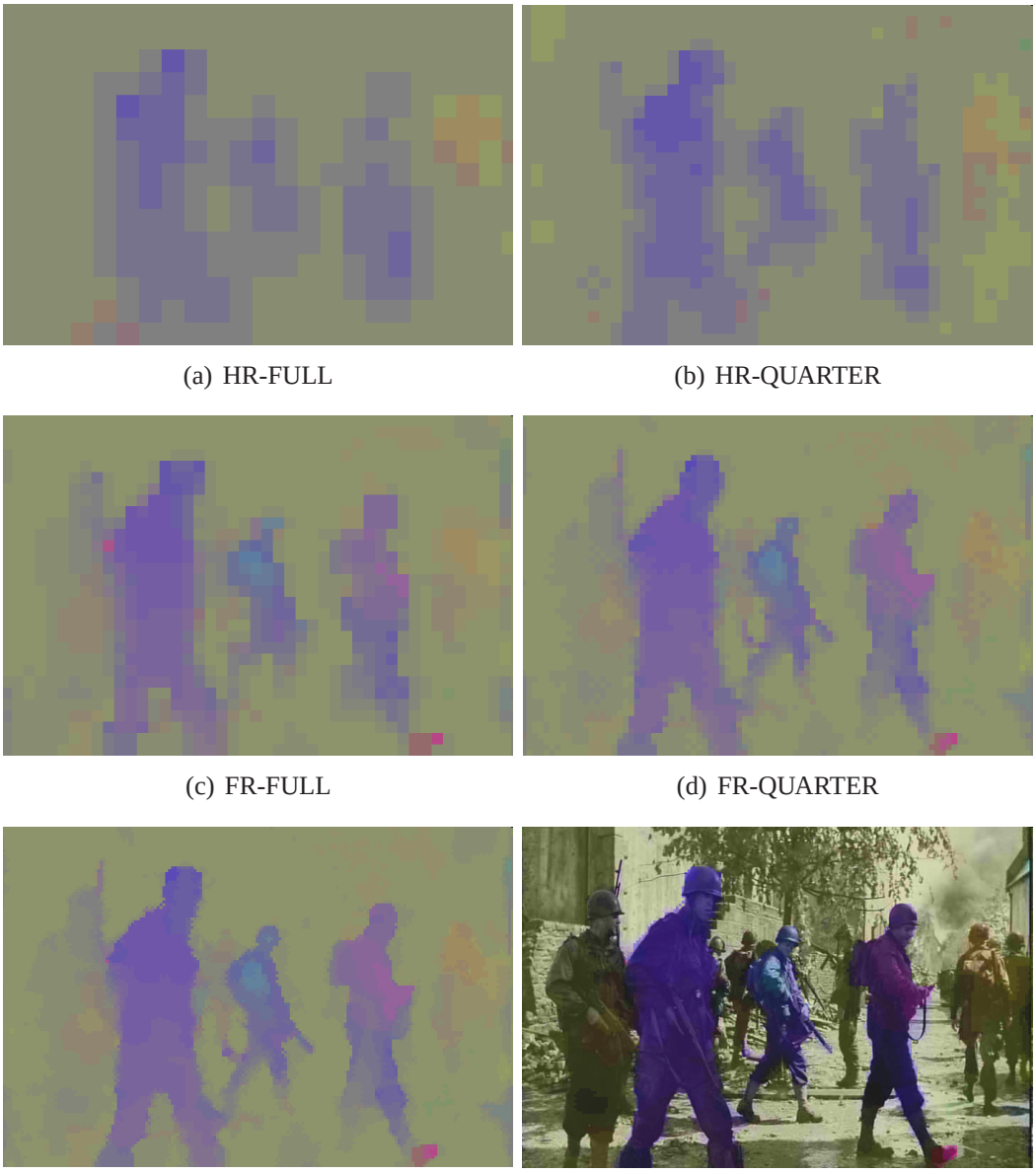
Energy minimization enables us to design an energy function that describes the desired properties of a solution that we are seeking for and minimize it to obtain the result. We designed the energy function by using the data and our prior knowledge of object motions. The data is evaluated by using the assumption of brightness constancy over a block of pixels together with the translational motion model. The prior knowledge is the smoothness of the motion-vector field, which is not valid across boundaries. The number of feasible solutions is exponentially large, and low-complexity approximation algorithms that converge to a desirable local minimum is needed. Unfortunately, the data term in the energy has many non-convexities in which the minimization algorithm can not break away from once attained.

To deal with the validity of the smoothness prior, we proposed to adjust the coefficients of the energy as we learn more about the motion via minimization iterations. By formulating the primal problem of deciding about a move to a new candidate motion vector as a quadratic program, we have shown that the dual problem provides the motion segmentation problem for that specific candidate motion-vector. By using a feasible solution of the dual problem, we have shown that one can update the energy function to reflect this new

information and therefore improve the smoothness term.

To achieve a low-complexity algorithm, we used standard-moves that use a local neighborhood. Standard-moves are hardware-friendly since only a single block can change its vector at a time. This requires less information exchange and fewer number of possible move combinations. We analyzed the performance of the minimization algorithm in terms of the convergence time to a local minimum, and the energy gap between the attained local minimum and the global minimum. We discussed how to choose the respective parameters for the best performance and quality.

To reduce the chances of getting stuck at a bad local minimum, we proposed to increase the overall weight of the smoothness cost, which is convex, as the iterations progress. Also, we discussed the importance of starting with a good initial motion-vector field by using non-spatial prior information.



(a) HR-FULL (b) HR-QUARTER

(c) FR-FULL (d) FR-QUARTER

(e) FR-QUARTER^2 (f) Gray-scale image with the color channels modulated by the motion information

Figure 7.3. A walk-through of the proposed algorithm using an image from the movie "Saving Private Ryan". With each new stage, motion boundaries are improved, and finer details like the rifle of the middle soldier are resolved.

(a) HR-QUARTER with $\lambda_2 = 12|BS|$

(b) HR-QUARTER with $\lambda_2 = 8|BS|$

(c) HR-QUARTER with $\lambda_2 = 4|BS|$

(d) HR-QUARTER with $\lambda_2 = 0.5|BS|$

Figure 7.4. The effect of different λ values demonstrated using an image taken from the movie "Fargo". Starting with large λ values prevents the single-walks breaking away from bad local minimums created by wrong motion-vectors of the initial field. In this example, repeating blinds create bad local minimums in the energy as in the green areas. Also, slippage of motion-vectors across strong one-directional intensity edges can create bad local minimums demonstrated by cyan areas. These bad local minimums can survive if λ is not chosen to be sufficiently small.

(a) FR-QUARTER^2 with $K = 64$

(b) FR-QUARTER^2 with $K = 4$

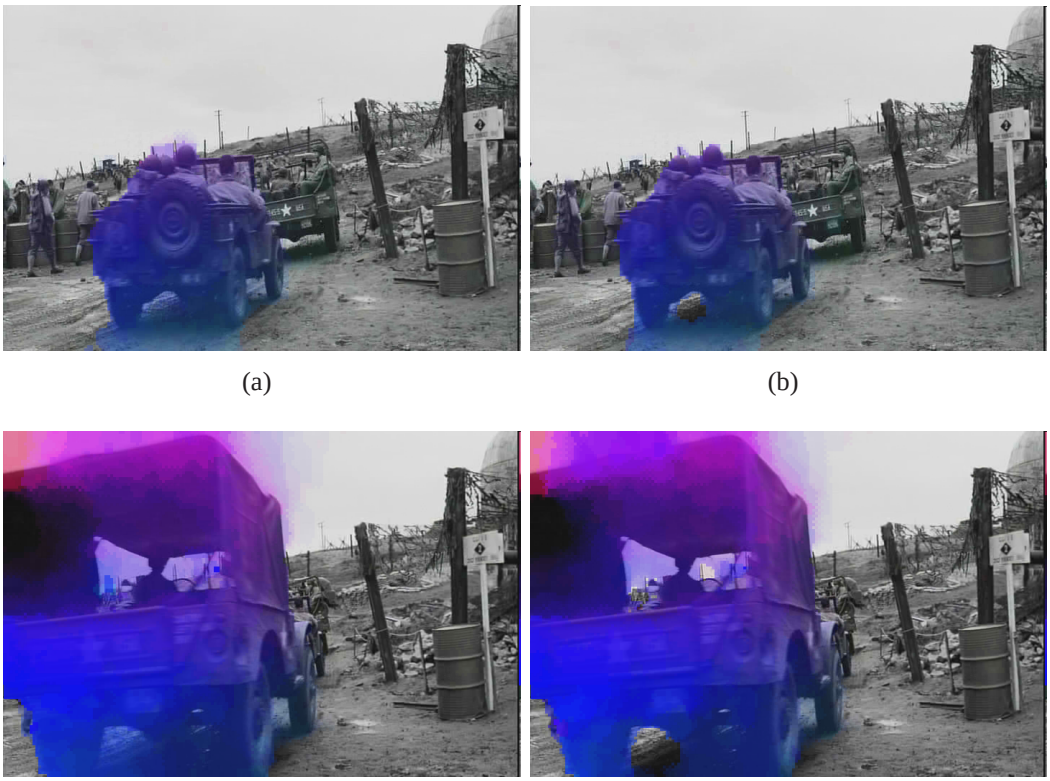
(c) FR-QUARTER^2 with $K = 1$

Figure 7.5. The effect of K values demonstrated using an image taken from the movie "Saving Private Ryan". With $K = 64$, the motion of the background seen through the jeep's glasses is merged with the jeep's motion. However, its motion-vectors are preserved with $K = 4$. Setting K to 1 preserves motion boundaries slightly better, but the motion-vector field becomes a bit noisy and some wrong vectors survive.

(a) $n_5 = n_4 = n_3 = n_2 = n_1 = 5$

(b) $n_5 = 9, n_4 = 8, n_3 = 5, n_2 = 4, n_1 = 5$

Figure 7.6. The change in quality by choosing varying number of iterations while still keeping the computational complexity constant. With more iterations in earlier stages of the algorithmic flow st the cost of less iterations on later stages, some bad local minimums can be overcome.



(a) (b)

(c) (d)

Figure 7.7. Analysis of $f(\hat{\beta}_{bi})$. Images on the left are produced with $f(\hat{\beta}_{bi}) = 1$, while the images on the right are produced by computing $\hat{\beta}_{bi}$ from (7.51) and choosing $f(\hat{\beta}_{bi}) = \left[\frac{C-\hat{\beta}_{bi}}{2C} + 0.5\right]_0^1$, where C is a constant.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 Contributions

In this thesis, low-complexity solutions to common video-processing problems are proposed.

Contrast enhancement is one of the most basic video applications in display products. We developed a histogram modification framework to be used in a histogram-equalization type approach to contrast enhancement. This framework equips us with tools to handle artifacts produced by histogram equalization. The complexity cost of solving each artifact is analyzed and finally an algorithm is presented.

We investigated the use of IIR filter for local contrast enhancement. Different types of IIR filters are investigated and the adaptivity of the delay coefficient is studied to prevent creating of artifacts around edges. Using these IIR filters, we designed skin-aware local contrast enhancement filters and ringing artifact reduction filters.

We studied the use of diffusion equations for image sharpening. Modification of these diffusion equations is studied to reduce noise boosting and creation of edge artifacts. A non-recursive solution that handles these issues is proposed.

Finally, we studied the motion estimation problem. We proposed methods to incorporate prior knowledge. Non-spatial prior knowledge is utilized using multi-hypothesis testing and Bayesian methods. Spatial prior knowledge is utilized by iteratively solving an energy-minimization problem. The primal-dual method is used to reliably impose the spatial priors. Simplifications for reducing computational complexity and heuristics for fast convergence of the minimization algorithm is discussed.

8.2 Future Research Directions

Our solutions include techniques such as global and local contrast enhancement, artifact reduction and sharpening that only use single frame information, and motion estimation that uses multi-frame information.

The proposed contrast enhancement approach can be improved by utilizing more criteria depending on the specific application such as mean-brightness conservation in the histogram modification framework. Some desired features in medical imaging can also be achieved using this framework for application specific improvements.

The diffusion based sharpening filter approach can be improved by adding the blocking artifact awareness on top of the ringing awareness we have presented.

Our multi-hypothesis testing framework for using non-spatial information is optimized for using previous frame's motion-vector field. This can be extended to using information from coarser resolutions in a hierarchical motion estimation.

In the study of energy-minimization for motion estimation we have used the spatial smoothness prior knowledge only. The smoothness prior does not work well in the presence of repeating patterns in the video: all the blocks/pixels in a neighborhood can consistently match with a false repetition. For these cases, projecting motion vectors to a different time instant can be used to create an extra prior. This will eliminate local minimums in the energy and help converging to a desired motion-vector field.

REFERENCES

[1] N. L. P. 1901-1921, *Ferdinand Braun Biography*. Amsterdam: Elsevier Publishing Company, 1967.

[2] *PCTechGuide: Cathode Ray Tubes*. http://www.pctechguide.com/42CRTMonitors.htm, 1990.

[3] J. Whitaker, *Standard Handbook of Video and Television Engineering*. McGraw-Hill Companies, 2003.

[4] G. Berbeccel, *Digital Image Display: Algorithms and Implementation*. England: John Wiley & Sons Ltd., 2003.

[5] G. de Haan, *Video processing for multimedia systems*. 2000.

[6] H.-D. Cheng and H. juan Xu, "A novel fuzzy logic approach to contrast enhancement," *Pattern Recognition*, vol. 33, pp. 809–819, May 2000.

[7] A. Beghdadi and A. L. Négrate, "Contrast enhancement technique based on local detection of edges," *Computer Vision, Graphics, and Image Processing*, vol. 46, pp. 162–174, May 1989.

[8] R. Sherrier and G. Johnson, "Regionally adaptive histogram equalization of the chest," vol. MI-6, no. 1, pp. 1 – 7, 1987.

[9] A. Polesel, G. Ramponi, and V. Mathews, "Image enhancement via adaptive unsharp masking," vol. 9, no. 3, pp. 505 – 10, 2000.

[10] A. V. Oppenheim, R. W. Schafer, and T. G. J. Stockham, "Nonlinear filtering of multiplied and convolved signals," *Proceedings of the IEEE*, vol. 56, pp. 1264–1291, Aug. 1968.

[11] Y. T. Kim, "Enhancement using brightness preserving bi-histogram equalization," *IEEE Transactions on Consumer Electronics*, vol. 43, pp. 1–8, February 1997.

[12] Y. Wang, Q. Chen, and B. Zhang, "Image enhancement based on equal area dualistic sub-image histogram equalization method," vol. 45, pp. 68–75, Feb. 1999.

[13] J. M. Gauch, "Investigations of image contrast space defined by variations on histogram equalization," in *CVGIP: Grap. Models Image Process.*, pp. 269–280, July 1992.

[14] J. A. Stark, "Adaptive image contrast enhancement usign generalizations of histogram equalization," *IEEE Transactions on Image Processing*, vol. 9, pp. 889–896, May 2000.

[15] Z. Y. Chen, B. R. Abidi, D. L. Page, and M. A. Abidi, "Gray-level grouping (GLG): An automatic method for optimized image contrast enhancement - part i: The basic method," *IEEE Transactions on Image Processing*, vol. 15, pp. 2290–2302, August 2006.

[16] Z. Y. Chen, B. R. Abidi, D. L. Page, and M. A. Abidi, "Gray-level grouping (GLG): An automatic method for optimized image contrast enhancement - part ii: The variations," *IEEE Transactions on Image Processing*, vol. 15, pp. 2303–2314, August 2006.

[17] J.-Y. Kim, L.-S. Kim, and S.-H. Hwang, "An advanced contrast enhancement using partially overlapped sub-block histogram equalization," vol. 11, pp. 475 – 484, Apr. 2001.

[18] Q. Wang and R. K. Ward, "Fast image/video contrast enhancement based on weighted thresholded histogram equalization," vol. 53, pp. 757–764, May 2007.

[19] S. Yang, J. H. Oh, and Y. Park, "Contrast enhancement using histogram equalization with bin underflow and bin overflow," in *Proceedings of ICIP 2003*, vol. vol.1, pp. 881–884, 2003.

[20] S.-D. Chen and A. Ramli, "Minimum mean brightness error bi-histogram equalization in contrast enhancement," vol. 49, no. 4, pp. 1310 – 19, 2003.

[21] C.-C. Sun, S.-J. Ruan, M.-C. Shie, and T.-W. Pai, "Dynamic contrast enhancement based on histogram specification," vol. 51, no. 4, pp. 1300 – 1305, 2005.

[22] S. Agaian, K. Panetta, and A. Grigoryan, "Transform-based image enhancement algorithms with performance measure," vol. 10, no. 3, pp. 367 – 82, 2001.

[23] S. Agaian, B. Silver, and K. Panetta, "Transform coefficient histogram-based image enhancement algorithms using contrast entropy," vol. 16, no. 3, pp. 741 – 58, 2007.

[24] S. Aghagolzadeh and C. Ersoy, "Transform image enhancement," *Optical Engineering*, vol. 31, no. 3, pp. 614 – 26, 1992.

[25] H. Greenspan, C. Anderson, and S. Akber, "Image enhancement by nonlinear extrapolation in frequency space," vol. 9, no. 6, pp. 1035 – 48, 2000.

[26] J. Tang, E. Peli, and S. Acton, "Image enhancement using a contrast measure in the compressed domain," *IEEE Signal Processing Letters*, vol. 10, no. 10, pp. 289 – 92, 2003.

[27] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 2nd ed., 2002.

[28] J. S. Lim, *Two-dimensional Signal and Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 1990.

[29] C. Fivez, P. Vuylsteke, P. Wambacq, P. Suetens, and E. Schoeters, "Multi-resolution contrast amplification in digital radiography with compensation for scattered radiation," p. 16A8, 1996.

[30] A. Laine, J. Fan, and W. Yang, "Wavelets for contrast enhancement of digital mammography," 1995.

[31] A. Laine, I. Koren, W. Yang, and F. Taylor, "A steerable dyadic wavelet transform and interval wavelets for enhancement of digital mammography," 1995.

[32] C. Fischer and R. A. Usmani, "Properties of some tridiagonal matrices and their application to boundary value problems," *SIAM Journal on Numerical Analysis*, vol. 6, pp. 127–142, 1969.

[33] S. M. Pizer, J. B. Zimmerman, and E. V. Staab, "Adaptive grey level assignment in ct scan display," *J. Comp. Assist. Tomogr.*, vol. 8, no. 2, pp. 300–305, 1984.

[34] P. M. N. nd R. C. Fitch, "Rel-time adaptive contrast enhancement," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI, no. 3, pp. 655–661, 1981.

[35] J. S. Lee, "Digital image enhancement and noise filtering by using local statistics," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI, no. 2, pp. 165–168, 1980.

[36] M. J. T. Smith and A. Docef, *A Study Guide for Digital Image Processing*. Georgia: Scientific Publishers, 1999.

[37] R. Deriche, "Fast algorithms for low-level vision.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 78–87, 1990.

[38] D. Dudgeon and R. Mersereau, "Multidimensional digital signal processing, prentice-hall, englewood cliffs," *NJ*, vol. 84, p. 1984.

[39] T. Arici and Y. Altunbasak, "Image local contrast enhancement using adaptive non-linear filters," in *IEEE International Conference on Image Processing (ICIP-06)*, pp. ??–??, To appear, IEEE, June 2006.

[40] O. V. Shentov, S. K. Mitra, and B. D. O. Anderson, "Stability testing of 2-d recursive digital filters based on a circuit-theoretic approach," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 3, pp. 387–392, 1989.

[41] J. H. Langlois and L. A. Roggman, "Attractive faces are only average," *Psychological Science*, vol. 1, pp. 115–121, 1990.

[42] T. Rzeszewski, "A novel automatic hue control system," *IEEE Transactions on Consumer Electronics*, vol. 21, pp. 155–162, 1975.

[43] L. Gu and D. Bone, "Skin color region detection in MPEG video sequences," in *10th International Conference on Image Analysis and Processing (ICIAP'99)*, pp. 898–903, 1999.

[44] S. L. Phung, D. Chai, and A. Bouzerdoum, "Adaptive skin segmentation in color images," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)*, vol. 3, pp. 353–356, Apr. 2003.

[45] J. B. Martinkauppi, M. N. Soriano, and M. V. Laaksonen, "Behavior of skin color under varying illumination seen by different cameras at different color spaces," in *Proc. SPIE ,Machine Vision Applications in Industrial Inspection IX*, vol. 4301, pp. 102–112, Apr. 2001.

[46] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," in *Proc. Graphicon*, pp. 85–92, Sept. 2003.

[47] C. Jeong, J. Kim, and K. Hong, "Appearance-based nude image detection," in *17th International Conference on Pattern Recognition (ICPR-04)*, vol. 4, pp. 467–470, Aug. 2004.

[48] A. Senior, R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 696–706, 2002.

[49] B. Menser and M. Wien, "Segmentation and tracking of facial regions in color image sequences," in *SPIE Visual Comm. and Image Processing (VCIP-00)*, vol. 4067, pp. 731–740, June 2000.

[50] J. Ahlberg, "A system for face localization and facial feature extraction." Report no.LiTH-ISY-R2172, Linkoping University.

[51] J. D. Hamilton, *Time Series Analysis*. Georgia: Princeton University Press, 1994.

[52] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Processing*, vol. 13, 2004.

[53] G.-H. Chen, C.-L. Yang, L.-M. Po, and S.-L. Xie, "Edge-based structural similarity for image quality assessment," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 933–936, May 2006.

[54] R. C. Gonzales and P. Wintz, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1977.

[55] G. Ramponi, "A cubic unsharp masking technique for contrast enhancement," *Signal Processing*, vol. 67, pp. 211–222, june 1998.

[56] G. Ramponi and A. Polesel, "A rational unsharp masking technique," 1998.

[57] G. Aubert, P. Kornprobst, and O. Faugeras, *Mathematical Problems in Image Processing: partial differential equations and the calculus of variation*. Springer, 2002.

[58] G. Gilboa, N. Sochen, and Y. Y. Zeevi, "Forward-and-backward diffusion processes for adaptive image enhancement and denoising," *IEEE Trans. on Image Processing*, vol. 11, pp. 689–703, July 2002.

[59] M. J. Black, G. Sapiro, D. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. on Image Processing*, vol. 7, no. 3, pp. 421–432, 1998.

[60] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[61] J. S. Lim, *Two dimensional signal and image processing*. New Jersey: Prentice-Hall, Inc., 1990.

[62] A. Beric, R. Sethuraman, J. L. van Meerbergen, and G. de Haan, "Memory-centric motion estimator.," *IEEE Conference on VLSI Design*, pp. 816–819, 2005.

[63] J. Konrad, *Motion detection and estimation in Handbook of Image and Video Processing*. Academic Press, 2005.

[64] G. de Haan and P. Biezen, "An efficient true-motion estimator using candidate vectors from a parametric motion model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 85–91, February 1998.

[65] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," vol. 6, pp. 721–741, November 1984.

[66] V. Kolmogorov and Y. Boykov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," in *EMMCVPR02*, p. 359 ff., 2002.

[67] Y. Boykov and O. Veksler, "Graph cuts in vision and graphics: Theories and applications," pp. 79–96, 2005.

[68] E.T.Jaynes, *Probability Theory: The Logic Of Science*. Cambridge University Press, 2003.

[69] B. Zeng, R. Li, and M. Liou, "Optimization of fast block motion estimation algorithms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 833–844, December 1997.

[70] R. Li, B. Zeng, and M. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, pp. 438–442, August 1994.

[71] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[72] B. K. P. Horn and B. Schunk, "Determining optical flow," vol. 17, pp. 185–204, 1981.

[73] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, p. 2001, 2001.

[74] A. Rav Acha, Y. Pritch, and S. Peleg, "Making a long video short: Dynamic video synopsis," pp. I: 435–441, 2006.

[75] A. Rav-Acha, P. Kohli, C. Rother, and A. Fitzgibbon, "Unwrap mosaics: A new representation for video editing," *ACM Transactions on Graphics*, vol. 27, pp. 17:1–17:11, Aug. 2008.

[76] O. V. P. D, "Efficient graph-based energy minimization methods in computer vision."

[77] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," vol. B-36, no. 2, pp. 192–236, 1974.

[78] V. Vazirani, *Approximation Algorithms*. Berlin: Springer-Verlag, 2001.

[79] *Edmonds-Karp Algorithm*. http://en.wikipedia.org/wiki/Edmonds-Karp_algorithm.